



TUGAS AKHIR - KS141501

PENGEMBANGAN SISTEM INFORMASI MONITORING PENGELOLAAN GEDUNG DI ITS

THE DEVELOPMENT OF BUILDING MONITORING AND MANAGEMENT INFORMATION SYSTEM IN ITS

THEO WIBISMANA SAKTI
NRP 05211140000065

Dosen Pembimbing I
Bekti Cahyo Hidayanto, S.Si., M.Kom.
Dosen Pembimbing II
Feby Artwodini Muqtadiroh, S.Kom., M.T

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



TUGAS AKHIR - KS141501

PENGEMBANGAN SISTEM INFORMASI MONITORING PENGELOLAAN GEDUNG DI ITS

THEO WIBISMANA SAKTI
NRP 05211140000065

Dosen Pembimbing I
Bekti Cahyo Hidayanto, S.Si., M.Kom.
Dosen Pembimbing II
Feby Artwodini Muqtadiroh, S.Kom., M.T

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

FINAL PROJECT - KS 141501

***THE DEVELOPMENT OF BUILDING MONITORING AND
MANAGEMENT INFORMATION SYSTEM IN ITS***

**THEO WIBISMANA SAKTI
NRP 05211140000065**

**Supervisor I
Bekti Cahyo Hidayanto, S.Si., M.Kom.
Supervisor II
Feby Artwodini Muqtadiroh, S.Kom., M.T**

**INFORMATION SYSTEMS DEPARTMENT
Faculty of Communication and Information Technology
Sepuluh Nopember Institut of Technology
Surabaya 2018**

LEMBAR PENGESAHAN
PENGEMBANGAN SISTEM INFORMASI
MONITORING PENGELOLAAN GEDUNG DI ITS
(SIMP G ITS)

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

THEO WIBISMANA SAKTI

NRP. 05211140000065

Surabaya, Juli 2018

KEPALA
DEPARTEMEN SISTEM INFORMASI

Dr. Ir. Aris Tjahyanto, M.Kom.

NIP.19650310 199102 1 001

LEMBAR PERSETUJUAN

PENGEMBANGAN SISTEM INFORMASI MONITORING PENGELOLAAN GEDUNG DI ITS (SIMPg ITS)

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

THEO WIBISMANA SAKTI

NRP. 05211140000065

Disetujui Tim Penguji: Tanggal Ujian : 23 Januari 2018
Periode Wisuda : September 2018

Bekti Cahyo Hidayanto, S.Si, M.Kom

(Pembimbing I)

Feby Artwodini, S.Kom, M.T

(Pembimbing 2)

Dr. Eng. Febriliyan Samopa, S.Kom, M.Kom (Penguji I)

Nisfu Asrul Sani, S.Kom, M.Sc

(Penguji II)

PENGEMBANGAN SISTEM INFORMASI MONITORING PENGELOLAAN GEDUNG DI ITS

Nama Mahasiswa : Theo Wibismana Sakti

NRP : 05211140000065

Jurusan : Sistem Informasi FTIf – ITS

Pembimbing 1 : Bkti Cahyo Hidayanto, S.Si., M.Kom.

Pembimbing 2 : Feby Artwodini Muqtadiroh, S.Kom., M.T

ABSTRAK

Institut Teknologi Sepuluh Nopember Surabaya (ITS) dalam masterplannya sudah mengatur rencana pengelolaan gedung sedemikian rupa agar sesuai dengan visi misi dan tujuan organisasinya. Namun, dengan luas lahan yang tergolong besar, terdapat kesulitan dalam memantau pembangunan dan perbaikan gedung seperti yang telah direncanakan. Hal ini menimbulkan beberapa permasalahan seperti kesulitan peninjauan gedung lama dan pertimbangan pembangunan gedung baru maupun pengalokasian prioritas dan anggaran perawatan gedung.

Pada kondisi saat ini peninjauan masih dilakukan dengan cara manual yakni melakukan peninjauan langsung dan pencatatan untuk kemudian diarsipkan oleh pihak pengelola gedung ITS. Masalah yang dialami oleh pihak pengelola gedung ITS dengan sistem ini adalah banyaknya data yang tidak terarsipkan dan hanya diketahui oleh pihak- pihak yang terlibat langsung pada saat peninjauan maupun pembangunan. Saat dilakukan pendataan ulang sering terjadi kendala dimana pihak yang terlibat sudah tidak menjabat sehingga data-data yang seharusnya diarsipkan pun hilang. Kesulitan lain yang dihadapi juga pada saat pihak pengelola gedung membutuhkan

pencarian data dengan cepat dan tepat, karena harus menghubungi pihak-pihak lain terlebih dahulu. Permasalahan inilah yang ingin dibantu oleh penelitian ini.

Penelitian ini membuat sebuah sistem informasi yang berdasarkan pada peta kondisi kekinian gedung ITS. Sistem informasi yang dihasilkan akan dapat menampilkan data-data terkait gedung dan bangunan yang sudah ada. Sistem juga dirancang untuk dapat membantu pengguna untuk mengajukan dan memantau perawatan gedung serta melihat data-data gedung terkait hubungannya dengan pengelola gedung. Sehingga sistem informasi diharapkan dapat membantu proses pemantauan dan renovasi gedung oleh pihak-pihak yang terlibat dalam pengelolaan bangunan.

Pada akhir pengembangan perangkat lunak, penulis juga melakukan pengujian untuk memastikan bahwa sistem yang dibangun telah sesuai dengan keperluan pengguna menggunakan metode blackbox testing.

Hasil dari tugas akhir ini adalah sebuah perangkat lunak yang dapat membantu pihak pengelola gedung di ITS dalam melakukan pengelolaan dan renovasi gedung.

Kata Kunci: software development, building management, monitoring

THE DEVELOPMENT OF BUILDING MONITORING AND MANAGEMENT INFORMATION SYSTEM IN ITS

Student Name : Theo Wibismana Sakti
NRP : 05211140000065
Department : Sistem Informasi FTIf – ITS
Supervisor 1 : Bkti Cahyo Hidayanto, S.Si., M.Kom.
Supervisor 2 : Feby Artwodini Muqtadiroh, S.Kom., M.T

ABSTRACT

In its masterplan, Institut Teknologi Sepuluh Nopember (ITS) have have devised plans to manage buildings and their respective use as to fulfill the organization's visions and missions. However, with it's vast land and a lot of buildings, difficulties arise when it is time to monitor and recheck the condition of said buildings overtime. This raises several issues such as the difficulty of reviewing the old buildings and the consideration of new building construction as well as the priority allocation and budget of the building maintenance.

In the current condition building reviewing is still done by human labour, that is doing direct observation and recording to then archived by the manager of ITS building. Problems experienced by the management of ITS building with this system are the amount of data that are not archived and only known by the parties involved directly at the time of review and development. When the data recollection is done, often times the parties involved are no longer serving so the data that should be archived was lost. Also, some problems encountered when the building manager needs to search for data quickly and

precisely, because they must contact the other parties first. These problems are what this research aims to help.

This research makes an information system based on the current condition of ITS's building. The resulting information system will be able to display data related to existing buildings. The system is also designed to help users to file and monitor building maintenance as well as review data relating to building managers. So, the information system is expected to help the process of monitoring and renovation of buildings by the parties involved in building management.

At the end of the software development, the authors also use black box testing to ensure that the system built is in accordance with the needs of users.

The result of this undergraduate thesis is a software that can help the building managers in ITS in managing and renovating existing buildings.

KATA PENGANTAR

Bismillahirrohmanirrohim

Puji Syukur Kepada Allah SWT karena atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan buku tugas akhir dengan judul

“PENGEMBANGAN SISTEM INFORMASI MONITORING PENGELOLAAN GEDUNG DI ITS (SIMPG ITS)”

yang merupakan satu syarat kelulusan pada Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Selama masa pengerjaan Tugas Akhir ini, penulis telah memperoleh banyak bantuan, bimbingan dan petunjuk dari berbagai pihak. Maka dari itu, dalam kesempatan ini penulis akan menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

- Allah SWT, yang telah memberikan kesehatan, kemudahan, kelancaran, dan kesempatan untuk penulis hingga dapat menyelesaikan Tugas Akhir ini.
- Kedua orangtua, adik, dan seluruh keluarga yang selalu hadir dan senantiasa mendoakan dan memberikan kasih sayang serta semangat tiada henti untuk menyelesaikan Tugas Akhir ini
- Bapak Dr. Ir. Aris Tjahyanto, M.Kom, selaku Ketua Departemen Sistem Informasi ITS, yang telah menyediakan fasilitas terbaik untuk kebutuhan penelitian mahasiswa.
- Rekan-rekan yang turut membantu penulis dalam menyelesaikan dan merapikan penulisan buku tugas akhir.

Penulis menyadari bahwa Tugas Akhir ini masih belum sempurna dan memiliki banyak kekurangan di dalamnya. Dan oleh karena itu, penulis meminta maaf atas segala kesalahan yang dibuat penulis dalam buku Tugas Akhir ini. Penulis membuka pintu selebar-lebarnya kepada pihak-pihak yang ingin memberikan kritik, saran, masukan, dan penelitian selanjutnya yang ingin menyempurnakan karya, dan Tugas Akhir ini. Semoga buku Tugas Akhir ini bermanfaat bagi seluruh pembaca.

Surabaya, Juli 2018

Theo W Sakti

DAFTAR ISI

LEMBAR PENGESAHAN.....	iii
LEMBAR PERSETUJUAN.....	iv
ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR.....	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah.....	1
1.2. Perumusan Masalah.....	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	3
1.6. Relevansi	4
BAB II TINJAUAN PUSTAKA.....	5
2.1. Studi Sebelumnya.....	5
2.2. Pengembangan Perangkat Lunak Prototyping	6
2.3. Black Box Testing.....	6
2.4. Hypertext Preprocessor (PHP), CodeIgniter (CI), dan Composer	7
2.5. PHPUnit dan ci-phpunit-test	8
2.6. PhpSpreadsheet	9
2.7. Proses Bisnis Renovasi Gedung ITS	9

2.8. Lahan ITS	10
BAB III METODOLOGI	13
3.1. Studi Literatur.....	15
3.2. Pengumpulan Kebutuhan dan Perancangan Perangkat Lunak.....	15
3.3. Pengembangan Perangkat Lunak	16
3.4. Uji Coba Perangkat Lunak	16
3.5. Penyusunan Dokumen Tugas Akhir	16
BAB IV PERANCANGAN	17
4.1. Tahap pengumpulan kebutuhan perangkat lunak	17
4.2. Tahap perancangan perangkat lunak	19
4.3. Tahap pengembangan perangkat lunak	43
4.4. Tahap uji coba perangkat lunak.....	44
BAB V IMPLEMENTASI	47
5.1. Purwarupa Pertama.....	47
5.1.1. Pengaturan framework <i>CodeIgniter</i>	48
5.1.2. Halaman utama aplikasi	49
5.1.3. Fungsi login.....	56
5.2. Purwarupa Kedua	58
5.2.1. Fungsi data gedung.....	61
5.2.2. Fungsi search	63
5.3. Purwarupa Ketiga	64
5.3.1. Fungsi renovasi gedung	66
5.3.2. Fungsi pekerjaan renovasi	71
5.4. Purwarupa Keempat	75

5.4.1. Fungsi data pengguna	75
5.5. Purwarupa Kelima.....	82
5.5.1. Fungsi <i>CUD</i> data gedung	82
5.5.2. Fungsi reset password pengguna	84
5.5.3. Fungsi unggah excel pekerjaan	87
BAB VI HASIL DAN PEMBAHASAN.....	91
6.1. Hasil dan Evaluasi Purwarupa Pertama.....	91
6.2. Hasil dan Evaluasi Purwarupa Kedua	92
6.3. Hasil dan Evaluasi Purwarupa Ketiga	94
6.4. Hasil dan Evaluasi Purwarupa Keempat	103
6.5. Hasil dan Evaluasi Purwarupa Kelima.....	105
BAB VII KESIMPULAN DAN SARAN	111
7.1. Kesimpulan.....	111
7.2. Saran.....	111
DAFTAR PUSTAKA	113
BIODATA PENULIS	115
LAMPIRAN A	1
LAMPIRAN B	1

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2.1 Proses Bisnis Renovasi Gedung di ITS	10
Gambar 2.2 Peta Masterplan ITS Tahun 2015	11
Gambar 3.1 Proses Pengerjaan Tugas Akhir.....	14
Gambar 4.1 Diagram Relasi Entitas Aplikasi SIMPG	19
Gambar 4.2 Diagram Swimlane Proses Bisnis Utama Renovasi Gedung di ITS	20
Gambar 4.3 Diagram Use Case Aplikasi.....	21
Gambar 4.4 Sequence Diagram Proses Login.....	39
Gambar 4.5 Sequence Diagram Proses Pengajuan Renovasi	40
Gambar 4.6 Sequence Diagram Proses Persetujuan/Penolakan Renovasi	41
Gambar 4.7 Sequence Diagram Proses Penambahan Pekerjaan Renovasi	42
Gambar 4.8 Sequence Diagram Proses Pengecekan Pekerjaan Renovasi	43
Gambar 5.1 Pengaturan Database <i>CodeIgniter</i>	48
Gambar 5.2 Pengaturan <i>Routing</i> untuk Memanggil Fungsi ...	49
Gambar 5.3 Fungsi Utama <i>Controller</i> Utama Beranda.....	51
Gambar 5.4 Fungsi Konstruktur Beranda.....	53
Gambar 5.5 Fungsi-Fungsi <i>Model</i> Halaman Utama.....	55
Gambar 5.6 Fungsi Login.....	57
Gambar 5.7 Diagram Relasi Entitas Aplikasi Revisi Pertama	59
Gambar 5.8 Fungsi Detail Data Gedung	61
Gambar 5.9 Fungsi Daftar Gedung	62
Gambar 5.10 Fungsi Search Nama Gedung	63
Gambar 5.11 Pengambilan Data Gedung dari Database	64
Gambar 5.12 Diagram Relasi Entitas Aplikasi Revisi Kedua	66
Gambar 5.13 Fungsi Pengambilan Data Renovasi	67
Gambar 5.14 Fungsi <i>create</i> Renovasi	68
Gambar 5.15 Fungsi <i>update</i> Renovasi	68
Gambar 5.16 Fungsi <i>delete</i> Renovasi.....	69
Gambar 5.17 Fungsi untuk Menampilkan Daftar Renovasi ...	70
Gambar 5.18 Fungsi <i>read</i> Data Pekerjaan.....	71
Gambar 5.19 Fungsi <i>delete</i> Pekerjaan.....	72
Gambar 5.20 Fungsi <i>update</i> Pekerjaan.....	72

Gambar 5.21 Fungsi <i>create</i> Pekerjaan.....	73
Gambar 5.22 Fungsi-fungsi Pengambilan Data dan Perubahan Data pada Database	74
Gambar 5.23 Fungsi Tampilan Utama <i>Management User</i>	76
Gambar 5.24 Fungsi <i>create</i> User Baru	77
Gambar 5.25 Fungsi <i>delete</i> User	78
Gambar 5.26 Fungsi <i>update</i> User	79
Gambar 5.27 Fungsi Utama pada <i>Manage_model</i>	80
Gambar 5.28 Fungsi-fungsi Penambahan Data User pada Database	80
Gambar 5.29 Fungsi Perubahan dan Penghapusan Data User	81
Gambar 5.30 Fungsi Menambah Data Gedung	83
Gambar 5.31 Fungsi Reset Password pada <i>controller</i> Manage	84
Gambar 5.32 Fungsi Reset Password pada <i>Manage_model</i> ...	84
Gambar 5.33 Fungsi Cek Password Default.....	85
Gambar 5.34 Fungsi Perubahan Password Pengguna	86
Gambar 5.35 Fungsi Baca File Excel	88
Gambar 5.36 Fungsi Model untuk Memasukkan Pekerjaan...	89
Gambar 6.1 <i>Code coverage</i> Hasil Unit Testing Purwarupa Pertama.....	91
Gambar 6.2 Lanjutan <i>Code coverage</i> Hasil Unit Testing Purwarupa Pertama.....	92
Gambar 6.3 Tampilan Utama Aplikasi pada Purwarupa Pertama	93
Gambar 6.4 Tampilan Utama Aplikasi Saat Pengguna Sudah Login	93
Gambar 6.5 Tampilan Akhir Halaman Utama Aplikasi	95
Gambar 6.6 Tampilan Data Lengkap Gedung.....	96
Gambar 6.7 Tampilan Login User.....	96
Gambar 6.8 Tampilan Login User Admin.....	97
Gambar 6.9 Tampilan Utama User Pegawai SIMRI	97
Gambar 6.10 Tampilan Utama User SARPRAS	98
Gambar 6.11 Tampilan Utama User WR II.....	99
Gambar 6.12 Tampilan Utama User Unit Fakultas/Jurusan...	99
Gambar 6.13 Tampilan Daftar Renovasi Gedung	100

Gambar 6.14 Tampilan Gedung Tanpa Renovasi	101
Gambar 6.15 Tampilan Pengajuan Proposal Renovasi	101
Gambar 6.16 Tampilan Detail Renovasi Termasuk Pekerjaan dan Proses Renovasi.....	102
Gambar 6.17 Tampilan Hapus Pekerjaan Renovasi	102
Gambar 6.18 Tampilan Mengubah Data Pekerjaan	103
Gambar 6.19 Tampilan <i>Management User</i>	104
Gambar 6.20 Tampilan Ubah Data Pengguna.....	104
Gambar 6.21 Tombol 'Tambah data gedung' pada Laman Utama	105
Gambar 6.22 Laman Form Penambahan Gedung Baru	106
Gambar 6.23 Tombol 'Ubah Data Gedung' pada Laman Detail Gedung	107
Gambar 6.24 Laman Form Mengubah Data Gedung	108
Gambar 6.25 Fungsi Reset Password Pengguna	108
Gambar 6.26 Notifikasi Password Reset.....	109
Gambar 6.27 Notifikasi pada Pengguna dengan Password Default.....	109
Gambar 6.28 Fungsi Unggah Excel Pekerjaan.....	110

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2.1 Daftar Penelitian Terkait	5
Tabel 4.1 Use Case Melihat Data Lengkap Gedung	21
Tabel 4.2 Use Case Mengubah Data Gedung	23
Tabel 4.3 Use Case Menambah Data Gedung Baru	24
Tabel 4.4 Use Case Menghapus Data Gedung	24
Tabel 4.5 Use Case Mengajukan Renovasi Gedung	25
Tabel 4.6 Use Case Menyetujui Renovasi Gedung	26
Tabel 4.7 Use Case Menolak Renovasi Gedung	27
Tabel 4.8 Use Case Menambah Pekerjaan Renovasi	28
Tabel 4.9 Use Case Mengubah Pekerjaan Renovasi	29
Tabel 4.10 Use Case Menghapus Pekerjaan Renovasi	30
Tabel 4.11 Use Case Mengecek Pekerjaan Renovasi	30
Tabel 4.12 Use Case Mengubah Renovasi Gedung	31
Tabel 4.13 Use Case Menghapus Data Renovasi Gedung	32
Tabel 4.14 Use Case Renovasi Gedung Selesai	33
Tabel 4.15 Use Case Menambah Data Pengguna	34
Tabel 4.16 Use Case Mengubah Data Pengguna	35
Tabel 4.17 Use Case Menghapus Data Pengguna	36
Tabel 4.18 Use Case Mengubah Password Pengguna	36
Tabel 4.19 Use Case Reset Password Pengguna	37

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

Bab pendahuluan ini menjelaskan latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan pengerjaan tugas akhir.

1.1. Latar Belakang Masalah

Dalam kondisi masa kini, banyak pihak dituntut untuk terus melakukan pelestarian alam dalam rangka mengatasi permasalahan yang berkaitan dengan keberlanjutan lingkungan, tidak terkecuali lembaga pendidikan perguruan tinggi. Institut Teknologi Sepuluh Nopember Surabaya (ITS) dalam masterplannya sudah menganut ideologi ramah lingkungan ini, termasuk dalam pembuatan ruang terbuka hijau (RTH) dan berbagai bangunan gedung lainnya.

Lahan ITS sendiri tergolong cukup luas, dengan luas lahan sebesar 1.720.000 m². Dari sekian lahan yang ada, terdapat lahan yang masih kosong dan lahan yang sudah dibangun gedung.[1] Luas lahan yang besar ini memberikan permasalahan tersendiri bagi pihak pengelola lahan ITS, yaitu dalam memantau pembangunan dan perbaikan gedung seperti yang telah direncanakan dalam masterplan ITS. Berdasarkan diskusi dengan pihak Sub Direktorat Implementasi Rencana Induk ITS (SIMRI), dalam kondisi sekarang belum terdapat peta lahan untuk menunjukkan keadaan lahan ITS yang belum maupun sudah dibangun sesuai pengamatan terbaru. ITS juga mengalami kesulitan untuk memantau lahan yang sekarang ini masih belum dimanfaatkan dan masih digunakan oleh pihak eco-campus maupun oleh masyarakat sekitar.

Kesulitan dalam memantau kondisi lahan dan gedung menimbulkan masalah tersendiri bagi pihak SIMRI untuk mengelola gedung yang ada. Terutama dalam pemantauan kondisi aktual gedung yang sudah dibangun dan kaitannya dengan perkembangan pembangunan sesuai dengan masterplan, yang merupakan tugas utama dari pihak SIMRI[2]. Belum

adanya sistem informasi terintegrasi untuk memantau keadaan gedung menambah beban SIMRI dalam meninjau kelayakan gedung lama dan mempertimbangkan perbaikan atau pembangunan gedung baru, terutama apabila terdapat pengajuan proposal oleh jurusan maupun fakultas. Hal ini juga mempengaruhi pihak sarana dan prasarana sebagai pelaksana perbaikan dan wakil rektor II (WR II) sebagai pembuat keputusan akhir dan pengalokasi anggaran pembangunan.

Disinilah penelitian ini mencoba membantu mengatasi masalah pemantauan gedung tersebut. Dengan membuat sebuah sistem informasi yang didasarkan pada peta kondisi kekinian lahan dan gedung ITS. Sistem informasi yang dibuat nantinya akan memuat beberapa fitur diantaranya membandingkan kondisi kekinian dengan masterplan ITS terbaru saat penelitian dibuat dan informasi lain terkait kondisi kekinian pembangunan ITS. Sistem informasi ini juga dirancang agar dapat dengan mudah diperbarui sehingga terus menerus menunjukkan kondisi kekinian lahan dan gedung ITS. Dalam pengembangan selanjutnya sistem informasi juga dapat digunakan untuk memantau kondisi bangunan yang sudah ada beserta pemeliharaan dan perawatan gedung secara berkala maupun bila terdapat kerusakan insidental.

1.2. Perumusan Masalah

Permasalahan yang dihadapi dalam tugas akhir ini adalah bagaimana sistem informasi monitoring yang sesuai dengan kebutuhan pihak manajemen ITS untuk memantau pengelolaan gedung di ITS?

1.3. Batasan Masalah

Batasan permasalahan dalam tugas akhir ini adalah sebagai berikut:

1. Fokus tugas akhir ini adalah melakukan pengembangan sistem informasi pengelolaan gedung dengan menggunakan

aplikasi berbasis web yang terintegrasi dengan sistem yang sudah ada di ITS.

2. Dalam konteks tugas akhir ini pengelolaan gedung yang dimaksud meliputi pemantauan dan renovasi gedung.

3. Aplikasi web yang dirancang difokuskan pada data lahan yang sudah tersedia pada arsip pengelola dan spesifikasi perangkat lunak yang disetujui bersama dengan pihak pengelola gedung di ITS.

4. Aplikasi memanfaatkan peta lahan yang telah tersedia pada penelitian sebelumnya dengan judul RANCANG BANGUN PETA INTERAKTIF TIGA DIMENSI MONITORING LAHAN (MONLAH) DI INSTITUT TEKNOLOGI SEPULUH NOPEMBER.

5. Pengguna aplikasi utama hanya meliputi perangkat kepegawaian ITS.

6. Studi kasus dilakukan di Institut Teknologi Sepuluh Nopember Surabaya pada kampus Sukolilo, kampus Manyar dan kampus Cokroaminoto tidak termasuk lingkup penelitian.

1.4. Tujuan Penelitian

Tujuan pengerjaan tugas akhir ini adalah untuk mengembangkan sebuah sistem informasi pengelolaan gedung untuk digunakan pada studi kasus di Institut Tekonologi Sepuluh Nopember Surabaya.

1.5. Manfaat Penelitian

Hasil tugas akhir ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Mempermudah pihak pengelola ITS dalam mengelola lahan dan bangunan dengan memanfaatkan sistem informasi.

2. Membantu memantau lahan ITS yang sudah memiliki bangunan gedung.

3. Membantu melakukan renovasi terhadap bangunan ITS yang sudah ada.

1.6. Relevansi

Tugas akhir ini relevan terhadap studi tahap sarjana dengan menggunakan ilmu rancang bangun aplikasi yang telah dipelajari pada perkuliahan untuk memecahkan masalah yang ada pada dunia nyata. Tugas akhir ini juga relevan dengan jurusan sistem informasi karena meliputi perancangan dan pembuatan sebuah sistem informasi dalam bentuk aplikasi web yang terintegrasi. Terakhir tugas akhir ini relevan dengan penjurusan Infrastruktur dan Keamanan Teknologi Informasi (IKTI), yaitu rancang bangun yang merupakan kompetensi IKTI dalam implementasi real time monitoring system pada roadmap lab IKTI.

BAB II

TINJAUAN PUSTAKA

Landasan teori ini berisi tentang literatur yang berhubungan dengan permasalahan tugas akhir. Untuk keperluan penelitian ini, studi penelitian sebelumnya difokuskan pada sistem informasi pengelolaan gedung dan pentingnya pembuatan sistem informasi monitoring bagi pihak pengelola.

2.1. Studi Sebelumnya

Sub-bab ini menjelaskan mengenai penelitian-penelitian sebelumnya yang memiliki relevansi dan/atau menjadi dasar dan bahan untuk penelitian ini.

Tabel 2.1 Daftar Penelitian Terkait

Judul paper 1	As-built data acquisition and its use in production monitoring and automated layout of civil infrastructure: A survey.
Penulis, tahun	Hyojoo Son, Frédéric Bosché, Changwan Kim; 2015
Deskripsi	Pengumpulan dan analisis data tiga dimensi dari bangunan gedung dalam rangka monitoring pembangunan dan operasional gedung.[3] Penelitian dilakukan untuk menganalisis infrastruktur gedung dan pengaplikasian data tersebut dalam proses pembangunan dan pengelolaan.
Keterkaitan	Penelitian mengenai monitoring gedung pada tahap pembangunan dan pengelolaan.
Judul paper 2	Real-time monitoring framework to investigate the environmental and structural performance of buildings.[4]
Penulis, tahun	Magdalena Hajdukiewicz, David Byrne, Marcus M. Keane, Jamie Goggins; 2015
Deskripsi	Membuat real-time monitoring framework dalam rangka memantau kinerja dan kesehatan gedung dari segi struktur dan pengaruhnya terhadap lingkungan. Menghasilkan informasi-informasi mengenai kinerja gedung

	dan memberikan kesempatan untuk strategi yang lebih baik dalam pengelolaan dan penggunaan gedung.
Keterkaitan	Penelitian mengenai monitoring gedung dalam kaitannya dengan penggunaan dan pengelolaan gedung dengan dampak lingkungan.

2.2. Pengembangan Perangkat Lunak Prototyping

Prototyping Software Development Life Cycle (SDLC – Siklus Hidup Pengembangan Perangkat Lunak) adalah sebuah metodologi dalam melakukan pengembangan perangkat lunak dengan menciptakan prototype/purwarupa perangkat lunak dalam beberapa iterasi pengembangan hingga perangkat lunak memenuhi spesifikasi perangkat lunak yang telah disetujui antara pengembang dan *stakeholder*[5]. Pengembangan perangkat lunak dengan metode prototyping menyediakan kerangka kerja pengembangan yang lebih fleksibel dan mudah dirubah dibandingkan dengan menggunakan metode pengembangan yang lebih tradisional seperti *waterfall*, *spiral*, dan sebagainya. Dalam penelitian ini, metode prototyping digunakan untuk mempermudah komunikasi dan pemenuhan kebutuhan pengguna karena fleksibilitas dan kemampuan metode ini dalam beradaptasi terhadap kebutuhan pengguna, utamanya kebutuhan fungsional yang dapat berubah seiring berjalannya pengembangan perangkat lunak[6].

2.3. Black Box Testing

Black box testing merupakan sebuah metode dalam melakukan pengujian perangkat lunak untuk memastikan bahwa perangkat lunak yang dikembangkan sesuai dengan keperluan pengguna. *Black box testing* selalu berdasarkan pada spesifikasi perangkat lunak, dan terkadang juga disebut *specification-based testing*. Bentuk paling sederhana dari *black box testing* adalah pengujian melakukan pemeriksaan pada spesifikasi perangkat lunak dan memilih uji kasus yang menggunakan fungsi – fungsi perangkat

lunak yang dapat dilihat secara external[7]. Terdapat beberapa jenis dalam *black box testing*, diantaranya *model-based testing*, *scenario-based testing*, *statistical testing*, dan *random testing*. Pada tugas akhir ini jenis pengujian yang dilakukan menggunakan *scenario-based testing*, atau pengujian berdasarkan skenario. Pengembang membuat skenario pengujian berdasarkan spesifikasi perangkat lunak yang telah disetujui dengan pengguna.

Black box testing dilakukan berdasarkan keperluan pengguna, sehingga keperluan yang tidak lengkap atau tidak terduga sebelumnya dapat dengan mudah diidentifikasi dan ditangani kemudian[8]. Pengujian juga dilakukan berdasarkan sudut pandang pengguna. Pengembang merasa pantas untuk menggunakan *black box testing* sebagai metode pengujian perangkat lunak, karena pada konteks tugas akhir ini tujuan pengembangan hanya untuk memenuhi keperluan pengguna pada perangkat lunak. Mempertimbangkan juga bahwa pengembang melakukan *prototyping* sebagai metode pengembangan. Pengujian aplikasi pada tiap akhir iterasi dapat dimanfaatkan sebagai sarana untuk meminta feedback pengguna dan mengidentifikasi keperluan pengguna yang belum ditemukan sebelumnya. Selain itu, pengguna tidak memberikan persyaratan optimasi kinerja aplikasi. Sehingga tidak diperlukan metode pengujian yang lebih rumit seperti *white box testing* untuk menguji perangkat lunak dari sudut pandang pengembang.

2.4. Hypertext Preprocessor (PHP), CodeIgniter (CI), dan Composer

PHP merupakan salah satu bahasa pemrograman yang dapat digunakan dalam membangun sebuah aplikasi berbasis web. Bahasa pemrograman PHP sangat sering digunakan pada berbagai aplikasi web di dunia dan merupakan salah satu dari 10 bahasa pemrograman yang dianggap paling matang.[9] Penggunaan bahasa pemrograman PHP dalam penelitian ini mempertimbangkan implementasi dan operasi aplikasi yang

mudah dan tidak menyusahkan pengguna. Dengan mengembangkan aplikasi berbasis web, pengguna akan lebih mudah untuk mengakses dan menggunakan aplikasi kapanpun dan dimanapun tersedia koneksi internet. Aplikasi juga mudah diakses baik menggunakan *browser* pada komputer ataupun menggunakan perangkat bergerak. Versi PHP yang digunakan pada penelitian ini adalah versi 7.0.22.

CodeIgniter merupakan *framework* aplikasi PHP untuk membuat bahasa pemrograman PHP mengikuti sistem *Object Oriented Programming* menggunakan *Model-View-Controller* (MVC). *Framework* ini termasuk sangat populer dan sering digunakan oleh pengembang web pada umumnya. Penggunaan CI dalam melakukan pengembangan aplikasi berbasis web ditujukan untuk mempercepat dan mempermudah proses penulisan kode, karena CI menyediakan beragam *library* untuk melakukan hal-hal yang biasa dilakukan menggunakan pemrograman PHP.[10] Versi terbaru yang juga digunakan penulis adalah versi 3.x.

Composer adalah *package manager* atau pengelola paket dan dependensi pada level aplikasi web untuk bahasa pemrograman PHP.[11] Composer memberikan standar untuk mengelola dependensi dan *library* yang diperlukan oleh suatu aplikasi PHP. Dalam konteks penelitian ini, composer digunakan untuk mengatur *library* PHPUnit dan *ci-phpunit-test* untuk pengujian aplikasi, dan ekstensi PHP lain seperti xDebug.

2.5. PHPUnit dan ci-phpunit-test

PHPUnit adalah *tools* atau perangkat dan kerangka kerja untuk melakukan unit testing pada bahasa pemrograman PHP. Unit testing yang dilakukan menggunakan PHPUnit lebih berorientasi terhadap pengembang perangkat lunak, karena pengujian yang dilakukan bersifat programming, atau menggunakan logika-logika programming untuk menentukan apakah perangkat lunak akan memberikan hasil yang diharapkan. Sebuah unit testing yang baik tidak hanya menguji apabila perangkat lunak akan memberikan hasil yang benar,

namun juga saat perangkat lunak seharusnya menghasilkan *error* atau kesalahan. PHPUnit sendiri merupakan salah satu bentuk dari arsitektur kerangka kerja unit testing xUnit.[12] Pada penelitian ini pengembang menggunakan PHPUnit versi 6 yang menyesuaikan versi PHP yang digunakan.

Sedangkan ci-phpunit-tests merupakan sebuah alat bantu untuk mempermudah penggunaan PHPUnit pada framework CI[13], dikarenakan CI memiliki fungsi-fungsi tambahan yang membuatnya berbeda dari implementasi pemrograman PHP pada umumnya. Sehingga PHPUnit perlu dirubah agar dapat melakukan unit testing pada aplikasi PHP berbasis *framework* CI.

2.6. PhpSpreadsheet

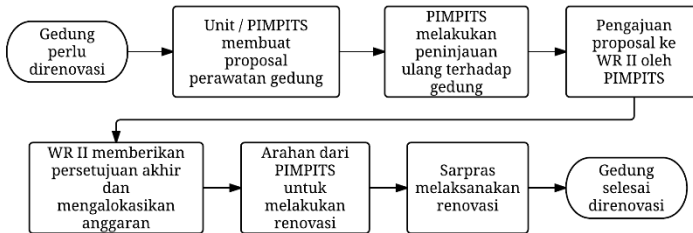
PhpSpreadsheet merupakan library php yang menyediakan seperangkat fungsi untuk membaca dan menulis ke bermacam format file spreadsheet seperti Excel atau LibreOffice Calc. PhpSpreadsheet merupakan perkembangan dari PHPExcel yang sekarang sudah tidak dikembangkan lagi.

2.7. Proses Bisnis Renovasi Gedung ITS

ITS melibatkan beberapa proses dalam melakukan pengelolaan gedungnya. Stakeholder utamanya adalah pihak SIMRI sebagai perencana pembangunan ITS dan WR II sebagai pihak yang mengetahui dan mengeluarkan kebijakan-kebijakan terkait prioritas pembangunan dan perbaikan gedung. Untuk pelaksanaan pembangunan dan perbaikannya sendiri akan dilakukan oleh bagian sarana dan prasarana ITS.

Sebagai contoh apabila unit jurusan mengajukan perbaikan gedung, maka SIMRI akan meninjau kondisi gedung dan menilai apakah perbaikan perlu dilakukan. Setelah pihak SIMRI menyetujui perbaikan gedung maka proposal diajukan ke WR II untuk persetujuan akhir dan pengalokasian prioritas dan anggaran dana. Setelah itu baru pihak sarana dan prasarana melakukan perbaikan gedung sesuai dengan arahan dari pihak

SIMRI. Untuk lebih jelas mengenai proses bisnis pembangunan dan renovasi gedung di ITS dapat dilihat pada proses flow berikut:



Gambar 2.1 Proses Bisnis Renovasi Gedung di ITS

Dari proses bisnis ini maka pihak SIMRI, WR II, sarana dan prasarana serta pihak lain seperti jurusan dan fakultas termasuk sebagai pengguna sistem dengan fungsi yang berbeda.

2.8. Lahan ITS

Berdasarkan data lahan, ITS memiliki lahan seluas 172 hektar[1] dimana lahan tersebut ada yang sudah dibangun gedung di atasnya dan yang belum atau sengaja tidak dibangun. Keseluruhan luas bangunan sendiri kurang lebih 150000 m². Lahan tersebut telah dikurangi oleh keluarnya PENS dan PPNS dari lingkup pengelolaan utama ITS.

Perencanaan pembangunan ITS selanjutnya telah diatur dalam sebuah masterplan pembangunan ITS. Dalam gedung rektorat ITS sudah ada maket perwakilan kondisi kekinian dan rencana pembangunan jangka pendek dari lahan ITS. Penelitian ini juga membantu dalam memantau lahan ITS dalam kondisi kekinian dan membandingkan dengan rencana pembangunan kedepan sesuai dengan masterplan yang ada.

Dalam masterplan ITS, terdapat banyak rencana pembangunan gedung yang saat ini masih belum ada dan merupakan lahan kosong. Untuk sementara ini, lahan tersebut dikelola oleh pihak

urban farming ITS dan masyarakat sekitar ITS. Denah pembangunan kedepan ITS dapat dilihat pada gambar berikut:



Gambar 2.2 Peta Masterplan ITS Tahun 2015

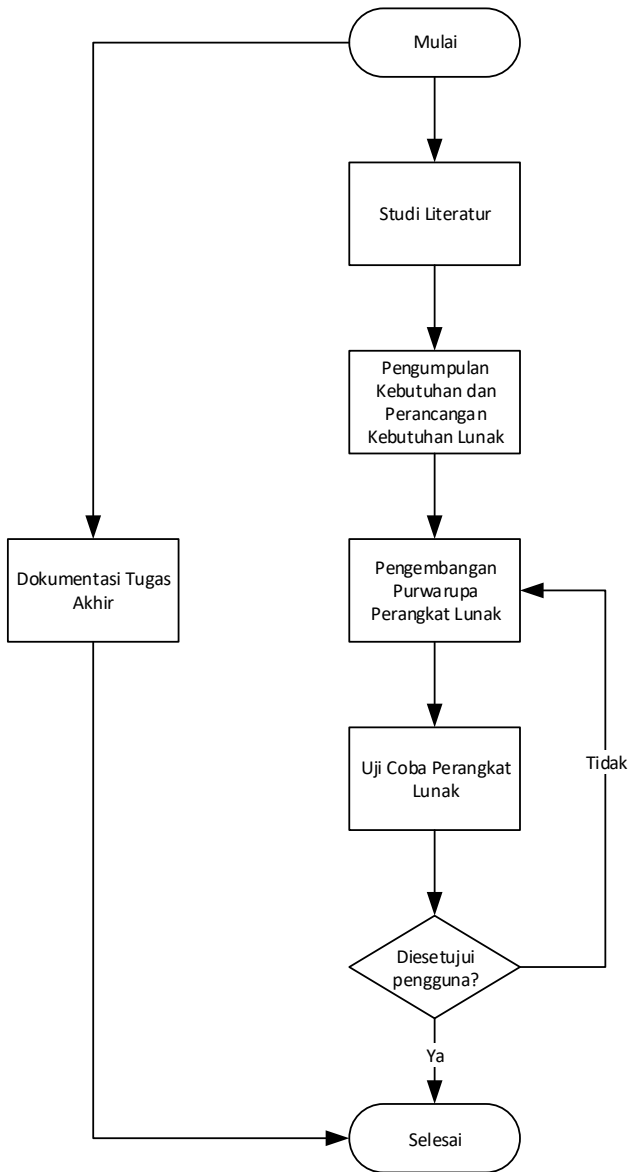
Perlu diingat bahwa rancangan masterplan ITS tersebut merupakan rancangan terakhir yang didapat pada saat pengumpulan data pada tahun 2016. Kedepannya masterplan ini

mungkin akan mengalami perubahan sesuai dengan perubahan kebijakan ITS. Oleh karena itu sesuai dengan hasil diskusi dengan pihak pembangunan dan perawatan gedung ITS maka sistem informasi akan membagi peta menjadi bagian-bagian kecil seluas 1,2 m² untuk mengakomodasi perubahan-perubahan yang mungkin terjadi dalam rencana pembangunan.

BAB III

METODOLOGI

Pada bagian ini akan dijelaskan mengenai proses pengerjaan tugas akhir. Metode yang digunakan bertujuan untuk menjadi pedoman agar pengerjaan tugas akhir dapat berlangsung secara sistematis. Untuk metode pengembangan sistem informasi pada tugas akhir ini akan digunakan model pengembangan perangkat lunak prototyping. Model pengembangan prototyping mampu memberikan sistem dengan fitur dasar kepada pengguna untuk keperluan percobaan sistem.[14] Dengan mengembangkan perangkat lunak dengan terus menerus mengambil pendapat/feedback dari pengguna, peneliti berharap sistem dapat menunjang pengelolaan gedung sesuai dengan yang dibutuhkan pengguna. Diagram alur pengerjaan tugas akhir ini dapat dilihat pada gambar berikut.



Gambar 3.1 Proses Pengerjaan Tugas Akhir

Tahapan pelaksanaan tugas akhir berisikan tentang tahapan-tahapan yang dilakukan untuk mengembangkan sistem informasi pada tugas akhir ini. Disini metode prototyping yang dilakukan masih mengikuti standar pengembangan perangkat lunak dimana masih terdapat tahapan pengumpulan kebutuhan dan perancangan, baru kemudian dilakukan iterasi pengembangan dan uji coba perangkat lunak. Metode prototyping ini terkadang juga disebut prototyping terstruktur[15]. Luaran yang diharapkan dari pengembangan perangkat lunak adalah sebuah sistem informasi terpadu berbasis peta yang dapat membantu pihak pengelola bangunan ITS untuk memantau perkembangan pembangunan dan perbaikan gedung ITS. Sistem informasi ini nantinya dapat menampilkan informasi status gedung, dengan datanya meliputi data luas gedung, data pemilik gedung, dan prosedur pengelolaan.

3.1. Studi Literatur

Tahapan studi literatur dilakukan untuk memahami konsep, metodologi, teori dan perangkat yang akan digunakan dalam pembuatan tugas akhir. Disini studi literatur difokuskan dalam mencari contoh sistem serupa dalam pembuatan sistem pengelolaan lahan. Pemahaman yang didapatkan antara lain mengenai rancangan awal sistem yang akan digunakan. Literatur-literatur yang digunakan didapatkan dari paper, buku, situs web, dan sumber lain seperti wawancara dengan pihak yang bersangkutan.

3.2. Pengumpulan Kebutuhan dan Perancangan Perangkat Lunak

Tahapan ini menjelaskan mengenai pengumpulan kebutuhan perangkat lunak dan membuat rancangan dasar perangkat lunak yang nantinya akan dibuat. Pengumpulan kebutuhan perangkat lunak utamanya didasarkan pada kebutuhan pihak SIMRI selaku pengelola pembangunan gedung. Rancangan perangkat lunak ini berdasarkan pada kebutuhan pihak pengelola lahan sesuai dengan yang telah disetujui sebelumnya.

3.3. Pengembangan Perangkat Lunak

Tahapan ini berisikan pengembangan perangkat lunak yang menjadi keluaran dari tugas akhir ini. Pelaksanaan tugas akhir akan kembali ke tahap ini sampai peneliti mendapat persetujuan dari stakeholder pada tahapan uji coba.

3.4. Uji Coba Perangkat Lunak

Tahap dilakukan bersamaan dengan tahap pengembangan perangkat lunak pada interval waktu tertentu. Perangkat lunak yang sudah jadi dipresentasikan kepada pihak pengelola lahan dan diuji coba untuk mengambil feedback dari pengguna. Tahapan ini juga menjadi parameter apakah perangkat lunak yang dibuat sudah sesuai dengan kebutuhan pengguna. Setelah tahap ini, pengembang akan kembali ke tahap pengembangan untuk merubah, menambah, atau mengurangi fitur perangkat lunak bila masih terdapat ketidaksesuaian dengan kebutuhan pengguna. Tahap uji coba akan selesai dilewati saat pihak stakeholder menyetujui hasil akhir perangkat lunak.

3.5. Penyusunan Dokumen Tugas Akhir

Tahap penyusunan dokumen tugas akhir merupakan tahap dalam menyusun buku tugas akhir ini sendiri. Tahapan dilakukan beriringan dengan tahapan lain pada pengerjaan tugas akhir termasuk perencanaan, pengembangan purwarupa dan uji coba perangkat lunak. Langkah ini diambil mempertimbangkan metode pengembangan prototyping dimana terdapat beberapa iterasi dalam melakukan pengembangan sehingga dokumen tugas akhir ini akan mengalami perubahan sesuai dengan luaran dari tahapan-tahapan yang mengalami iterasi tersebut.

BAB IV

PERANCANGAN

Pada bab ini akan dijelaskan mengenai proses perancangan perangkat lunak berdasarkan metode pengembangan prototyping yang digunakan. Bab ini meliputi perancangan perangkat lunak sesuai dengan metode yang telah dijelaskan pada bab sebelumnya pada tiap tahapannya.

4.1. Tahap pengumpulan kebutuhan perangkat lunak

Pada tahap ini perancangan difokuskan untuk mencari kebutuhan dasar dari perangkat lunak secara menyeluruh. Untuk itu akan dilakukan wawancara dengan pihak Sub Direktorat Implementasi Rencana Induk (SIMRI) selaku pengguna perangkat lunak nantinya. Beberapa pertanyaan yang diajukan untuk menentukan kebutuhan utama perangkat lunak antara lain:

1. Proses bisnis pengguna untuk menentukan alur perangkat lunak
2. Pihak mana saja yang nantinya menggunakan perangkat lunak
3. Fungsi apa saja yang diperlukan dari perangkat lunak
4. Antar muka yang diharapkan oleh pengguna sehingga perangkat lunak mudah untuk digunakan
5. Luaran yang dihasilkan pada tahap ini berupa fungsi utama perangkat lunak dan diagram alur proses bisnis pengguna.

Setelah melakukan *requirement gathering*, pengembang membuat daftar *functional requirement* (FR) perangkat lunak sebagai dasar pengembangan perangkat lunak. FR yang berhasil didapat dari tanya jawab dengan pengguna termasuk:

1. Aplikasi dapat menampilkan peta seluruh ITS beserta bangunan di atasnya, dimana peta ITS menjadi dasar acuan gedung

2. Aplikasi dapat menampilkan data-data terkait gedung seperti luas bangunan, jumlah lantai, dan sebagainya
3. Aplikasi dapat menampilkan progress renovasi gedung pada tiap renovasi dan pekerjaan
4. Aplikasi dapat mengomunikasikan persetujuan akhir untuk melaksanakan renovasi gedung dan alokasi dana antar stakeholder
5. Aplikasi dapat mengkalkulasi dan menampilkan luas lahan terbangun secara keseluruhan

Pada tahap pengumpulan kebutuhan awal ini juga didapat peran-peran pengguna dalam proses bisnis renovasi gedung, sehingga peran pengguna dalam sistem juga disesuaikan. Peran pengguna dalam sistem dibagi menjadi 5 tingkat dan dijelaskan beserta hak aksesnya seperti berikut:

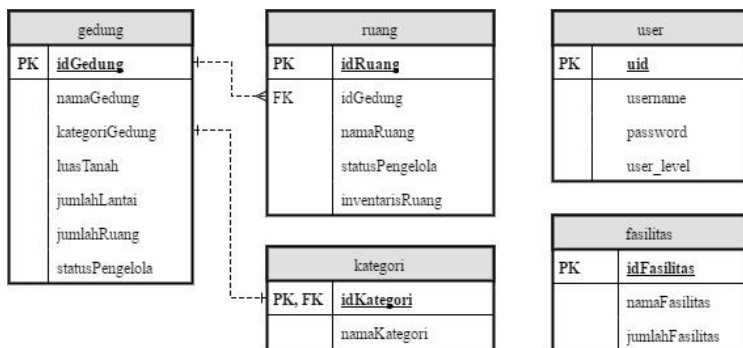
1. Administrator (superuser): memiliki hak akses utama untuk melakukan semua fungsi pada sistem.
2. Pegawai SIMRI: memiliki hak akses untuk melihat, mengajukan dan mengubah data renovasi, menentukan dan mengubah pekerjaan yang perlu dilakukan pada saat renovasi, dan menilai apabila renovasi telah cukup dan selesai dilakukan.
3. Wakil Rektor II: memiliki hak akses untuk melihat data renovasi yang diajukan dan menyetujui atau menolak renovasi beserta menentukan anggaran renovasi bila disetujui. Pengguna juga dapat melihat proses pengerjaan renovasi namun tidak dapat mengubah atau memberikan pekerjaan tambahan.
4. SARPRAS: selaku pihak pelaku renovasi, memiliki hak akses untuk melihat data renovasi dan pekerjaan, dan melakukan konfirmasi apabila pekerjaan telah selesai dilakukan.
5. Unit Fakultas: memiliki hak akses untuk melihat data renovasi dan pekerjaan beserta prosesnya, dan melakukan pengajuan renovasi baru.

Pengguna Unit Fakultas juga dibagi hak aksesnya berdasarkan fakultas sehingga hanya dapat melakukan akses pada data terkait fakultas masing-masing. Fakultas dibagi menjadi 5, yaitu FTSP, FTI, FMIPA, FTK, dan FTIf. Semua pengguna memiliki hak akses untuk melihat data-data terkait gedung, termasuk pengguna yang tidak melakukan login pada sistem.

4.2. Tahap perancangan perangkat lunak

Tahap ini utamanya berfokus pada perancangan antar muka dan desain awal perangkat lunak sebelum masuk pada proses coding. Perancangan perangkat lunak mencakup pembuatan diagram use case dan swimlane proses bisnis seperti yang dikemukakan oleh pengguna. Luaran yang dihasilkan antara lain desain antar muka perangkat lunak dan diagram alur perangkat lunak.

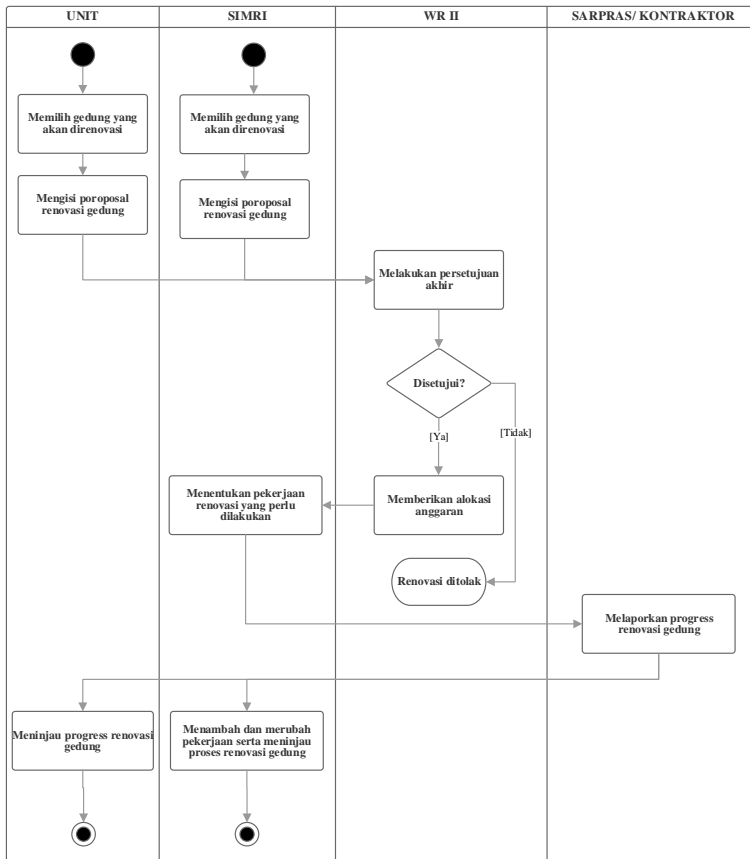
Perancangan awal skema database sebagai dasar perangkat lunak juga dilakukan pada tahap ini. Rancangan awal database aplikasi yang digunakan dijelaskan pada diagram *Entity Relationship Diagram* (ERD) berikut:



Gambar 4.1 Diagram Relasi Entitas Aplikasi SIMPG

Database yang digunakan menggunakan MySQL versi 10.1.26-MariaDB yang merupakan database standar pengembangan aplikasi web menggunakan XAMPP pada platform windows.

Alur perangkat lunak secara umum dijelaskan pada diagram swimlane berikut:

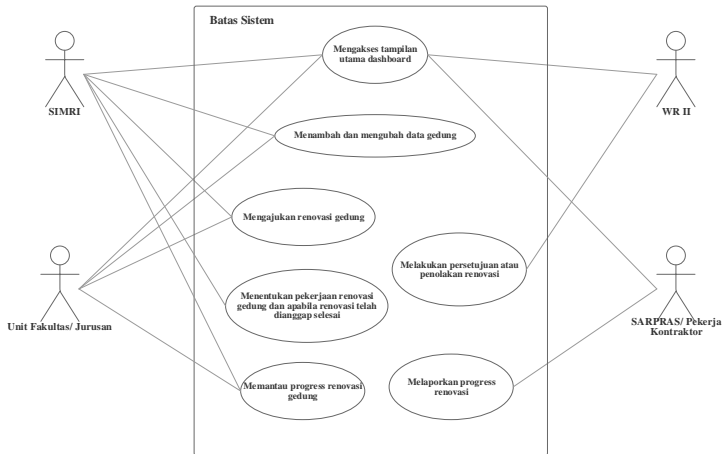


Gambar 4.2 Diagram Swimlane Proses Bisnis Utama Renovasi Gedung di ITS

Diagram swimlane menunjukkan proses bisnis renovasi gedung di ITS yang dilakukan melalui bantuan sistem perangkat lunak. Untuk pengajuan renovasi dapat dilakukan oleh pihak SIMRI maupun pihak unit fakultas/ jurusan. Antar muka perangkat lunak dirancang sesuai dengan kebutuhan pengguna yaitu untuk membantu monitoring gedung yang ada. Berdasarkan

kebutuhan tersebut antar muka dirancang sedemikian rupa agar mudah digunakan namun tidak menghambat fungsi utama.

Alur perangkat lunak diatas mengikuti proses bisnis pengguna dalam melakukan renovasi gedung di ITS dan interaksi pengguna dengan perangkat lunak. Lebih lengkapnya dapat dilihat pada diagram *use case* dibawah:



Gambar 4.3 Diagram Use Case Aplikasi

Dari diagram *use case* tersebut pengguna juga membuat *scenario use case* sistem untuk tiap fungsi yang ada dan dijabarkan sebagai berikut:

Tabel 4.1 Use Case Melihat Data Lengkap Gedung

Use Case Melihat Data Lengkap Gedung	
Tujuan	Pengguna dapat melihat data gedung secara rinci
Aktor	Admin, Pegawai SIMRI, WR II, SARPRAS, Unit/Fakultas
Skenario Utama	
Kondisi Awal	-
Aksi Aktor	Reaksi Sistem

1. Pengguna memilih gedung dari daftar nama gedung	2. Sistem menampilkan pop-up data gedung
Skenario Alternatif 1	
Kondisi Awal	-
Aksi Aktor	Reaksi Sistem
1. Pengguna memilih gedung menggunakan pin pada tampilan peta	2. Sistem menampilkan nama dan luas gedung dalam bentuk pop-up
3. Pengguna melakukan klik pada link 'Data Lengkap'	4. Sistem menampilkan data lengkap gedung pada tab browser baru
Skenario Alternatif 2	
Kondisi Awal	-
Aksi Aktor	Reaksi Sistem
1. Pengguna mengisikan nama gedung pada kolom pencarian dan klik 'Cari'	2. Sistem mencari nama gedung sesuai dengan masukan pengguna
	3. Sistem menampilkan daftar nama gedung
4. Pengguna memilih nama gedung dari daftar nama gedung	5. Sistem menampilkan pop-up data gedung
Skenario Alternatif 3	
Kondisi Awal	-
Aksi Aktor	Reaksi Sistem
1. Pengguna mengisikan nama gedung pada kolom pencarian dan klik 'Cari'	2. Sistem mencari nama gedung sesuai dengan masukan pengguna
	3. Sistem menampilkan daftar nama gedung
4. Pengguna memilih nama gedung dari pin gedung pada peta	5. Sistem menampilkan pop-up nama dan luas gedung
6. Pengguna melakukan klik pada link 'Data Lengkap'	7. Sistem menampilkan data lengkap gedung pada tab baru

Kondisi Akhir	Sistem menampilkan data lengkap gedung yang dipilih
----------------------	---

Tabel 4.2 Use Case Mengubah Data Gedung

Use Case Mengubah Data Gedung	
Tujuan	Pengguna dapat mengubah data gedung yang sudah ada
Aktor	Admin, Pegawai SIMRI
Skenario Utama	
Kondisi Awal	Pengguna telah melakukan login dengan hak akses yang memadai
Aksi Aktor	Reaksi Sistem
1. Pengguna memilih gedung dari list nama gedung	2. Sistem menampilkan pop-up data gedung
3. Pengguna melakukan klik tombol 'Ubah'	4. Sistem menampilkan data gedung dalam bentuk form
5. Pengguna melakukan perubahan data	
6. Pengguna klik tombol 'Simpan'	7. Sistem melakukan validasi masukan pengguna dan menampilkan error pada data yang tidak valid
	8. Sistem melakukan update database gedung
	9. Redirect ke laman gedung dan menampilkan notifikasi bahwa data gedung telah berhasil diubah
Skenario Alternatif	
Kondisi Akhir	Sistem menampilkan data rinci gedung setelah dilakukan perubahan

Tabel 4.3 Use Case Menambah Data Gedung Baru

Use Case Menambah Data Gedung Baru	
Tujuan	Pengguna dapat menambahkan data gedung baru
Aktor	Admin, Pegawai SIMRI
Skenario Utama	
Kondisi Awal	Pengguna telah melakukan login dengan hak akses yang memadai
Aksi Aktor	Reaksi Sistem
1. Pengguna melakukan klik tombol ‘Tambah’	2. Sistem menampilkan form data gedung
3. Pengguna mengisikan data gedung	
4. Klik ‘Simpan’	5. Sistem melakukan validasi data gedung dan menampilkan error apabila ada
	6. Sistem menambahkan data gedung pada database gedung
	7. Redirect ke laman utama dan menampilkan notifikasi bahwa data gedung berhasil ditambahkan
Kondisi Akhir	Sistem menampilkan laman utama aplikasi

Tabel 4.4 Use Case Menghapus Data Gedung

Use Case Menghapus Data Gedung	
Tujuan	Pengguna dapat menghapus data gedung yang ada

Aktor	Admin, Pegawai SIMRI
Skenario Utama	
Kondisi Awal	Pengguna telah melakukan login dengan hak akses yang memadai
Aksi Aktor	Reaksi Sistem
1. Pengguna memilih gedung dari daftar nama gedung	2. Sistem menampilkan pop-up data gedung
3. Klik tombol 'Hapus'	4. Sistem menampilkan pop-up warning untuk menghapus data gedung
5. Klik 'Ya'	6. Sistem menghapus data gedung dari database gedung
	7. Redirect ke laman utama dan menampilkan notifikasi bahwa data gedung berhasil dihapus
Kondisi Akhir	Sistem menampilkan laman utama aplikasi

Tabel 4.5 Use Case Mengajukan Renovasi Gedung

Use Case Mengajukan Renovasi Gedung	
Tujuan	Pengguna dapat mengajukan renovasi gedung
Aktor	Pegawai SIMRI, Unit/Fakultas

Skenario Utama	
Kondisi Awal	Pengguna telah melakukan login dengan hak akses yang memadai
Aksi Aktor	Reaksi Sistem
1. Pengguna memilih gedung dari daftar nama gedung	2. Sistem menampilkan pop-up data gedung
3. Klik tombol ‘Ajukan renovasi’	4. Sistem menampilkan form pengajuan renovasi
5. Pengguna mengisi data alasan renovasi	6. Sistem melakukan validasi data renovasi dan mengisi tanggal pengajuan
	7. Sistem memasukkan data renovasi gedung ke database renovasi
Kondisi Akhir	Sistem menampilkan laman renovasi gedung

Tabel 4.6 Use Case Menyetujui Renovasi Gedung

Use Case Menyetujui Renovasi Gedung	
Tujuan	Pengguna dapat menyetujui renovasi Gedung dan menentukan anggaran renovasi
Aktor	WR II
Skenario Utama	
Kondisi Awal	Pengguna telah melakukan login dengan hak akses yang memadai
Aksi Aktor	Reaksi Sistem
1. Pengguna memilih judul renovasi dari daftar renovasi yang diajukan	2. Sistem menampilkan data renovasi
3. Klik tombol ‘Setujui Renovasi’	4. Sistem menampilkan pop-up anggaran renovasi

5. Pengguna mengisi jumlah anggaran	6. Sistem melakukan validasi dan mengubah status renovasi menjadi disetujui, serta menambahkan data anggaran ke database renovasi
Kondisi Akhir	Sistem menampilkan laman renovasi gedung

Tabel 4.7 Use Case Menolak Renovasi Gedung

Use Case Menolak Renovasi Gedung	
Tujuan	Pengguna dapat menolak renovasi gedung
Aktor	WR II
Skenario Utama	
Kondisi Awal	Pengguna telah melakukan login dengan hak akses yang memadai
Aksi Aktor	Reaksi Sistem
1. Pengguna memilih judul renovasi dari daftar renovasi yang diajukan	2. Sistem menampilkan data renovasi
3. Klik tombol 'Tolak renovasi'	4. Sistem menampilkan pop-up alasan penolakan
5. Pengguna mengisi alasan renovasi ditolak (opsional)	6. Sistem melakukan validasi alasan penolakan dan mengubah status renovasi menjadi ditolak, serta menambahkan alasan penolakan jika diberikan oleh pengguna
Kondisi Akhir	Sistem menampilkan laman renovasi gedung

Tabel 4.8 Use Case Menambah Pekerjaan Renovasi

Use Case Menambah Pekerjaan Renovasi	
Tujuan	Pengguna dapat menambahkan pekerjaan renovasi yang perlu dilakukan
Aktor	Pegawai SIMRI
Skenario Utama	
Kondisi Awal	Pengguna telah melakukan login dengan hak akses yang memadai
Aksi Aktor	Reaksi Sistem
1. Pengguna memilih judul renovasi dari daftar renovasi yang diajukan	2. Sistem menampilkan data renovasi
3. Klik tombol 'Tambah Pekerjaan'	4. Sistem menampilkan form menambah pekerjaan
5. Pengguna mengisikan data pekerjaan	6. Sistem menambahkan data pekerjaan renovasi ke database pekerjaan renovasi
Skenario Alternatif	
Kondisi Awal	Pengguna telah melakukan login dengan hak akses yang memadai
Aksi Aktor	Reaksi Sistem
1. Pengguna memilih judul renovasi dari daftar renovasi yang diajukan	2. Sistem menampilkan data renovasi
3. Klik tombol 'Unggah Pekerjaan'	4. Sistem menampilkan form untuk mengunggah file excel berisi daftar pekerjaan renovasi
5. Pengguna mengunggah file daftar pekerjaan	6. Sistem menyimpan file excel dan memproses daftar pekerjaan ke

	format data pekerjaan pada database
	7. Sistem menambahkan data pekerjaan renovasi ke database pekerjaan renovasi
Kondisi Akhir	Sistem menampilkan laman renovasi gedung dengan daftar pekerjaan renovasi

Tabel 4.9 Use Case Mengubah Pekerjaan Renovasi

Use Case Mengubah Pekerjaan Renovasi	
Tujuan	Pengguna dapat mengubah pekerjaan renovasi gedung
Aktor	Pegawai SIMRI
Skenario Utama	
Kondisi Awal	<ul style="list-style-type: none"> - Pengguna telah melakukan login dengan hak akses yang memadai - Pengguna telah membuka laman detail renovasi
Aksi Aktor	Reaksi Sistem
1. Pengguna melakukan klik tombol 'Ubah' pada pekerjaan dari daftar pekerjaan renovasi	2. Sistem menampilkan form mengubah pekerjaan
3. Pengguna memasukkan data pekerjaan	
4. Klik tombol 'Submit'	5. Sistem melakukan validasi masukan pengguna dan mengubah data pekerjaan pada database pekerjaan

Kondisi Akhir	Sistem menampilkan laman renovasi gedung dengan daftar pekerjaan yang telah diubah
----------------------	--

Tabel 4.10 Use Case Menghapus Pekerjaan Renovasi

Use Case Menghapus Pekerjaan Renovasi	
Tujuan	Pengguna dapat menghapus pekerjaan renovasi
Aktor	Pegawai SIMRI
Skenario Utama	
Kondisi Awal	<ul style="list-style-type: none">- Pengguna telah melakukan login dengan hak akses yang memadai- Pengguna telah membuka laman detail renovasi
Aksi Aktor	Reaksi Sistem
1. Pengguna melakukan klik tombol ‘Hapus’ pada pekerjaan dari daftar pekerjaan renovasi	2. Sistem menampilkan pop-up warning untuk menghapus pekerjaan
3. Klik ‘Hapus’	4. Sistem menghapus data pekerjaan dari database pekerjaan renovasi dan menampilkan notifikasi bahwa pekerjaan telah dihapus
Kondisi Akhir	Sistem menampilkan laman renovasi gedung dengan daftar pekerjaan yang telah diubah

Tabel 4.11 Use Case Mengecek Pekerjaan Renovasi

Use Case Mengecek Pekerjaan Renovasi

Tujuan	Pengguna dapat melakukan pengecekan pada pekerjaan renovasi yang sudah dilakukan
Aktor	SARPRAS
Skenario Utama	
Kondisi Awal	<ul style="list-style-type: none"> - Pengguna telah melakukan login sebagai peran SARPRAS - Pengguna telah membuka laman detail renovasi
Aksi Aktor	Reaksi Sistem
1. Pengguna memberikan tanda centang pada daftar pekerjaan mana saja yang telah dilakukan	2. Sistem mengubah status pekerjaan menjadi telah dilakukan
	3. Redirect ke laman detil renovasi gedung
Kondisi Akhir	Sistem menampilkan laman detil renovasi gedung dengan daftar pekerjaan

Tabel 4.12 Use Case Mengubah Renovasi Gedung

Use Case Mengubah Renovasi Gedung	
Tujuan	Pengguna dapat mengubah data renovasi gedung
Aktor	Admin, Pegawai SIMRI
Skenario Utama	
Kondisi Awal	<ul style="list-style-type: none"> - Pengguna telah melakukan login dengan hak akses yang memadai - Pengguna telah membuka laman renovasi gedung atau laman daftar seluruh renovasi

Aksi Aktor	Reaksi Sistem
1. Pengguna melakukan klik tombol 'Ubah Data Renovasi' dari daftar renovasi	2. Sistem menampilkan form perubahan data renovasi
3. Pengguna memasukkan memperbarui data renovasi	
4. Klik tombol 'Submit'	5. Sistem melakukan validasi data renovasi dan mengisi tanggal perubahan data renovasi
	6. Sistem mengubah data renovasi gedung ke database renovasi
Kondisi Akhir	Sistem menampilkan laman daftar renovasi gedung

Tabel 4.13 Use Case Menghapus Data Renovasi Gedung

Use Case Menghapus Data Renovasi Gedung	
Tujuan	Pengguna dapat menghapus data renovasi gedung
Aktor	Admin, Pegawai SIMRI
Skenario Utama	
Kondisi Awal	<ul style="list-style-type: none"> - Pengguna telah melakukan login dengan hak akses yang memadai - Pengguna telah membuka laman renovasi gedung atau laman daftar seluruh renovasi
Aksi Aktor	Reaksi Sistem

1. Pengguna melakukan klik tombol ‘Hapus’ dari daftar renovasi	2. Sistem menampilkan pop-up warning untuk menghapus data renovasi
3. Klik ‘Hapus’	4. Sistem menghapus data renovasi dari database
Kondisi Akhir	Sistem menampilkan laman daftar renovasi gedung

Tabel 4.14 Use Case Renovasi Gedung Selesai

Use Case Renovasi Gedung Selesai	
Tujuan	Pengguna dapat menandai bahwa renovasi yang dilakukan sudah selesai
Aktor	Admin, Pegawai SIMRI
Skenario Utama	
Kondisi Awal	<ul style="list-style-type: none"> - Pengguna telah melakukan login dengan hak akses yang memadai - Pengguna telah membuka laman renovasi gedung atau laman daftar seluruh renovasi
Aksi Aktor	Reaksi Sistem
1. Pengguna melakukan klik tombol ‘Selesai Renovasi’ dari daftar renovasi	2. Sistem mengubah status renovasi menjadi selesai dan memasukkan tanggal hari ini sebagai tanggal selesai renovasi ke database renovasi
	3. Redirect ke tampilan sebelumnya, dapat berarti laman daftar renovasi gedung atau laman daftar seluruh renovasi

Kondisi Akhir	Sistem menampilkan laman daftar renovasi
----------------------	--

Tabel 4.15 Use Case Menambah Data Pengguna

Use Case Menambah Data Pengguna	
Tujuan	Admin dapat menambahkan pengguna baru ke dalam sistem
Aktor	Admin
Skenario Utama	
Kondisi Awal	Admin telah melakukan login dan valid
Aksi Aktor	Reaksi Sistem
1. Admin memilih 'Tambah User' pada dropdown di menu bar aplikasi	2. Sistem menampilkan form penambahan pengguna baru
3. Admin memasukkan data pengguna	4. Sistem melakukan validasi masukan pengguna dan menambahkan data pengguna ke database
	5. Sistem menampilkan notifikasi bahwa pengguna baru telah didaftarkan
Skenario Alternatif	
Kondisi Awal	Admin telah melakukan login dan valid
Aksi Aktor	Reaksi Sistem
1. Admin memilih 'Manage User' pada dropdown di menu bar aplikasi	2. Sistem menampilkan laman daftar pengguna
3. Admin klik tombol 'Add User'	4. Sistem menampilkan form penambahan pengguna
5. Admin memasukkan data pengguna baru	6. Sistem melakukan validasi masukan

	pengguna dan menambahkan data pengguna ke database
	7. Sistem menampilkan notifikasi bahwa pengguna baru telah didaftarkan
Kondisi Akhir	Sistem menampilkan laman daftar pengguna

Tabel 4.16 Use Case Mengubah Data Pengguna

Use Case Mengubah Data Pengguna	
Tujuan	Admin dapat menambahkan pengguna baru ke dalam sistem
Aktor	Admin
Skenario Utama	
Kondisi Awal	<ul style="list-style-type: none"> - Admin telah melakukan login dan valid - Admin berada pada laman daftar pengguna
Aksi Aktor	Reaksi Sistem
1. Admin memilih 'Update User' dari daftar pengguna	2. Sistem mengambil data pengguna dari database dan menampilkan form perubahan data pengguna
3. Admin melakukan perubahan data	4. Sistem melakukan validasi masukan pengguna dan memperbarui data pengguna pada database
	5. Sistem menampilkan notifikasi bahwa data pengguna telah dirubah
Kondisi Akhir	Sistem menampilkan laman daftar pengguna

Tabel 4.17 Use Case Menghapus Data Pengguna

Use Case Menghapus Data Pengguna	
Tujuan	Admin dapat menghapus data pengguna pada sistem
Aktor	Admin
Skenario Utama	
Kondisi Awal	<ul style="list-style-type: none"> - Admin telah melakukan login dan valid - Admin berada pada laman daftar pengguna
Aksi Aktor	Reaksi Sistem
1. Admin memilih 'Delete User' dari daftar pengguna	2. Sistem menampilkan pop-up warning menghapus data pengguna
3. Admin memilih 'Hapus'	4. Sistem menghapus data pengguna terpilih dari database
	5. Sistem menampilkan notifikasi bahwa data pengguna telah dihapus
Kondisi Akhir	Sistem menampilkan laman daftar pengguna

Tabel 4.18 Use Case Mengubah Password Pengguna

Use Case Mengubah Password Pengguna	
Tujuan	Pengguna dapat melakukan perubahan password
Aktor	Admin, Pegawai SIMRI, WR II, SARPRAS, Unit/Fakultas

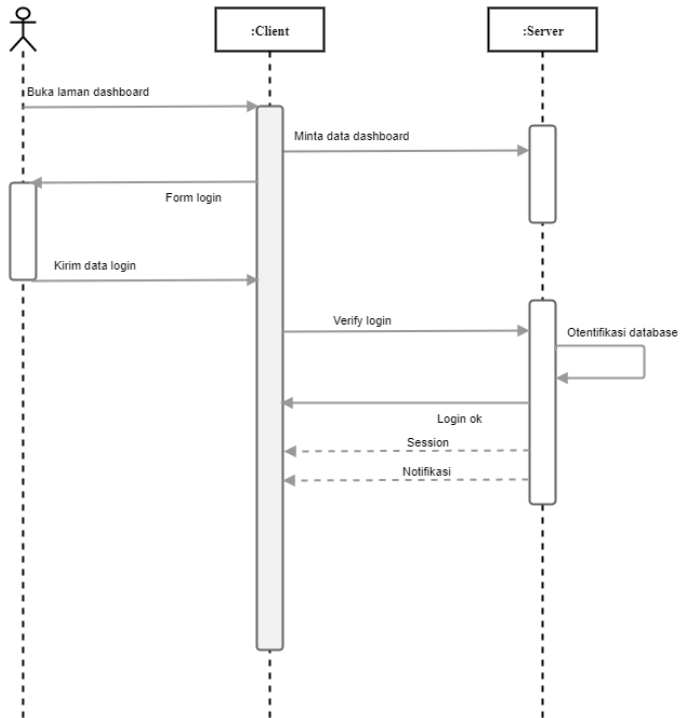
Skenario Utama	
Kondisi Awal	Pengguna telah melakukan login
Aksi Aktor	Reaksi Sistem
6. Pengguna melakukan klik tombol 'Ubah Password' pada navigation bar	7. Sistem menampilkan pop-up perubahan password pengguna
8. Pengguna memasukkan password baru	9. Sistem melakukan update password pengguna pada database
	10. Sistem menampilkan notifikasi bahwa password pengguna telah diubah
Kondisi Akhir	Sistem menampilkan laman utama aplikasi

Tabel 4.19 Use Case Reset Password Pengguna

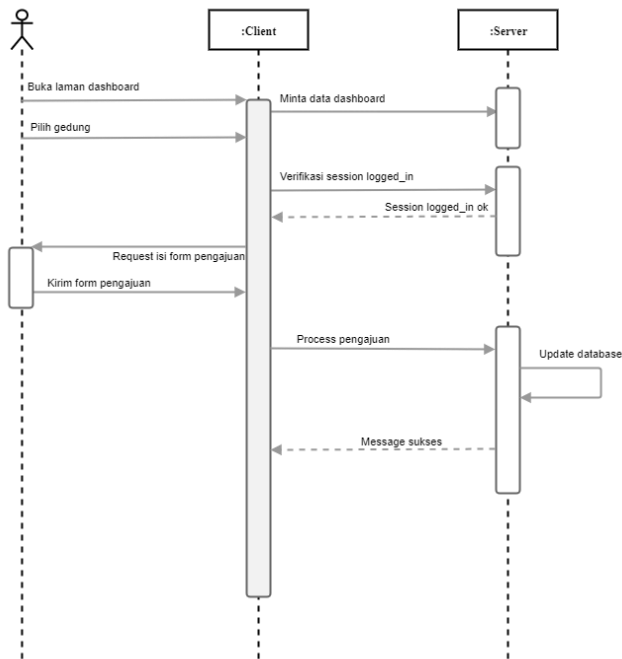
Use Case Reset Password Pengguna	
Tujuan	Admin dapat melakukan reset password apabila pengguna lain lupa dengan passwordnya
Aktor	Admin
Skenario Utama	

Kondisi Awal	<ul style="list-style-type: none"> - Admin telah melakukan login dan valid - Admin berada pada laman daftar pengguna
Aksi Aktor	Reaksi Sistem
1. Admin melakukan klik pada tombol 'Reset Password' dari daftar pengguna yang terdaftar pada sistem	2. Sistem menampilkan pop-up warning reset password pengguna
3. Klik 'Reset Password'	4. Sistem akan mengubah password pengguna pada database
	5. Sistem menampilkan notifikasi bahwa password pengguna telah dikembalikan ke password default '123qwe'
Kondisi Akhir	Sistem menampilkan laman daftar pengguna

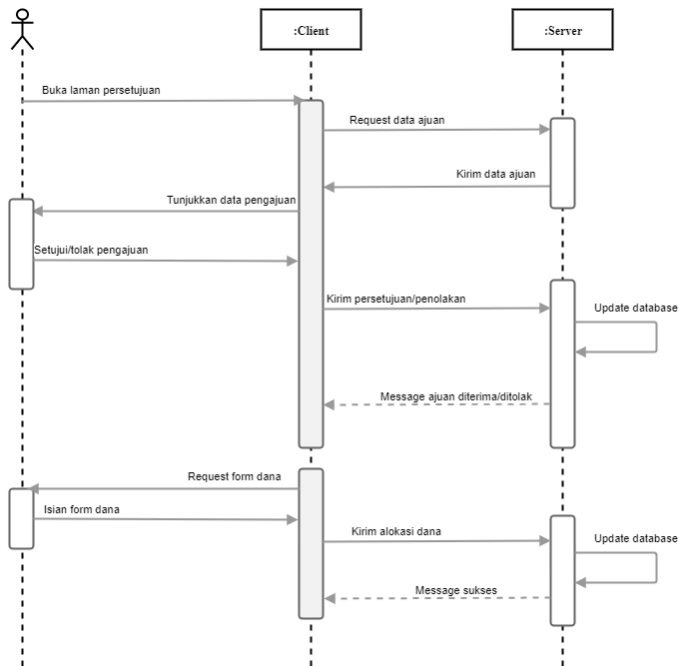
Kemudian penulis juga membuat *sequence diagram* aplikasi untuk mempermudah proses pengembangan purwarupa aplikasi yang dijelaskan pada diagram-diagram berikut:



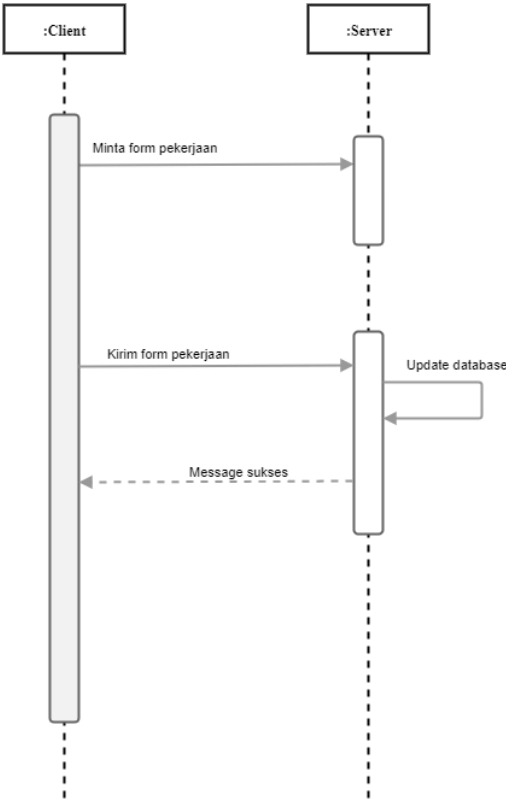
Gambar 4.4 Sequence Diagram Proses Login



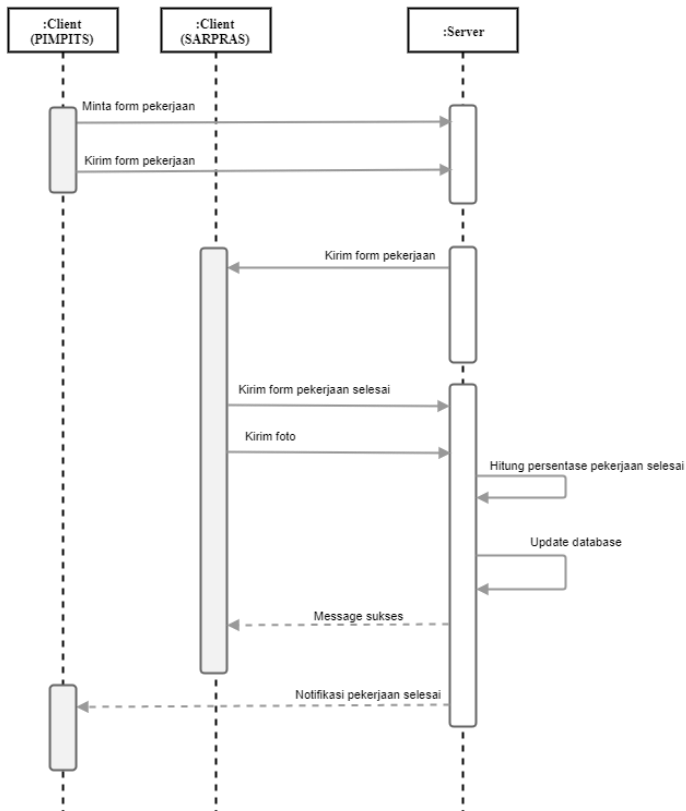
Gambar 4.5 Sequence Diagram Proses Pengajuan Renovasi



Gambar 4.6 Sequence Diagram Proses Persetujuan/Penolakan Renovasi



Gambar 4.7 Sequence Diagram Proses Penambahan Pekerjaan Renovasi



Gambar 4.8 Sequence Diagram Proses Pengecekan Pekerjaan Renovasi

4.3. Tahap pengembangan perangkat lunak

Pada tahap pengembangan dilakukan penulisan kode perangkat lunak berdasarkan rancangan yang telah dibuat. Karena menggunakan metode prototyping maka tahap pengembangan ini dilakukan dalam beberapa iterasi/pengulangan pembuatan purwarupa. Fase purwarupa yang dibuat adalah sebagai berikut:

1. Purwarupa pertama: Pengembangan fungsi dasar dan antar muka awal dashboard utama

2. Purwarupa kedua: Pengembangan fungsi overview keseluruhan gedung dan menampilkan data-data terkait gedung yang sudah ada
3. Purwarupa ketiga: Pengembangan fungsi pengajuan dan peninjauan renovasi gedung

Luaran yang dihasilkan pada tahap pengembangan berupa purwarupa perangkat lunak yang memiliki fungsi sesuai dengan iterasi pengembangannya. Sehingga pada iterasi pertama purwarupa yang dihasilkan akan memiliki fungsi dasar dashboard utama dan desain antar muka awal. Pada iterasi-iterasi selanjutnya tahap ini nantinya termasuk melakukan perubahan iterasi sebelumnya apabila diminta oleh pengguna pada tahap evaluasi.

4.4. Tahap uji coba perangkat lunak

Pada tahap ini purwarupa perangkat lunak yang sudah jadi dipresentasikan kepada pihak stakeholder untuk mendapatkan persetujuan dan umpan balik dari pengguna. Persetujuan pengguna digunakan sebagai dasar bahwa fungsi yang disediakan oleh perangkat lunak sudah memenuhi kebutuhan pengguna. Dan umpan balik yang didapat digunakan untuk memperbaiki perangkat lunak dari segi fungsi apabila masih belum mencakup kebutuhan pengguna atau antar muka perangkat lunak agar lebih mudah digunakan oleh pengguna. Persetujuan dan umpan balik dicatat dan ditandatangani oleh pengguna dalam form evaluasi pengguna. Tahap ini akan dilakukan beberapa kali sesuai dengan fase pengembangan purwarupa perangkat lunak.

Untuk mendapat persetujuan dan umpan balik dari pengguna maka pada tahap ini juga dilakukan *blackbox testing* terhadap purwarupa sesuai fase pengembangannya. Testing yang dilakukan bersifat *specification-based* sehingga pengguna dapat mengecek apakah fungsi yang disediakan perangkat lunak sudah sesuai harapan pengguna dan memenuhi kebutuhan perangkat lunak yang telah disepakati. Umpan balik yang

didapat dari pengguna pada tahap ini juga diolah sebagai masukan pada saat melakukan pengembangan purwarupa perangkat lunak pada iterasi selanjutnya.

Dokumen terkait tahap evaluasi perangkat lunak berupa checklist dan use case evaluasi perangkat lunak pada tiap iterasi dapat dilihat pada bagian lampiran. Selain itu juga terdapat dokumen persetujuan ketua SIMRI untuk kelanjutan pengembangan tiap iterasi.

Halaman ini sengaja dikosongkan

BAB V

IMPLEMENTASI

Pada tahap implementasi dilakukan pengembangan perangkat lunak dalam bentuk purwarupa perangkat lunak. Pada bab ini juga akan dijelaskan mengenai mengenai tahap pengembangan perangkat lunak berupa *coding* beserta penjelasan fungsi-fungsi sesuai dengan *feature requirement* yang telah diterangkan pada bab sebelumnya. Menggunakan metode *prototyping*, pengembangan dilakukan dalam beberapa kali iterasi dan menghasilkan purwarupa aplikasi pada tiap iterasinya. Karena pengembang menggunakan framework CI yang menerapkan aplikasi berbasis objek sehingga penjelasan kode pada tiap iterasi berdasarkan pada sistem *model-view-controller* (MVC).

5.1. Purwarupa Pertama

Pada iterasi pertama dilakukan pengembangan terhadap fungsi-fungsi dasar pendukung dan fungsi CRUD database. Dengan menggunakan framework CI maka fungsi ini berupa kode pada bagian model dan controller.

Menggunakan rancangan awal database perangkat lunak maka dilakukan pengembangan terhadap beberapa *functional requirement* sistem yaitu:

1. Sistem dapat mengambil data dan melakukan login *user* sesuai peran pengguna dalam sistem
2. Sistem dapat melakukan fungsi *logout*
3. Sistem dapat menampilkan halaman utama berisi peta ITS secara keseluruhan
4. Sedangkan *non-functional requirement* yang dikembangkan pada iterasi pertama ini terkait keamanan aplikasi dengan menggunakan fungsi *session* CI sebagai dasar fungsi-fungsi lain pada aplikasi. Pembuatan kode pada iterasi pertama ini dijelaskan sebagai berikut:

5.1.1. Pengaturan framework *CodeIgniter*

Sebelum mengembangkan *controller*, *model*, dan *view* aplikasi, pengembang terlebih dahulu melakukan pengaturan awal untuk menggunakan PHP dengan framework CI.

```
$query_builder = TRUE;

$db['default'] = array(
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => 'zeruel13',
    'database' => 'siplah',
    'dbdriver' => 'mysqli',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
);
```

Gambar 5.1 Pengaturan Database *CodeIgniter*

Tangkapan layar menunjukkan pengaturan database aplikasi, menggunakan database mysql tabel siplah dengan login database root dan password ‘zeruel13’. Database mysql berada pada server yang sama dengan aplikasi web, sehingga pengaturan memanggil database melalui *localhost*, kemudian terdapat beberapa pengaturan lain seperti *character set* dan *db collation* yang disamakan dengan pengaturan database mysql. Disini pengembang juga mengaktifkan fitur *query_builder* yang disediakan oleh framework CI untuk mengakses database.

```

$route['default_controller'] = 'beranda';
$route['404_override'] = '';
$route['translate_uri_dashes'] = FALSE;
$route['login'] = 'beranda/masuk';
$route['logout'] = 'beranda/keluar';
$route['gedung/(:num)'] = 'beranda/detailGedung/$1';
$route['gedung/edit(:num)'] = 'beranda/ubahGedung/$1';
$route['gedung/tambah'] = 'beranda/tambahGedung';
$route['renovasi/(:any)'] = 'beranda/dataRenovasi/$1';
$route['renovasi/pekerjaan/(:num)'] =
'beranda/listPekerjaan/$1';
$route['search'] = 'beranda/searchGedung';
$route['renovasi/pekerjaan/baru'] = 'beranda/tambahPekerjaan';
$route['renovasi/pekerjaan/edit/(:any)'] =
'beranda/ubahPekerjaan/$1';
$route['renovasi/del/(:num)'] = 'beranda/hapusRenovasi/$1';
$route['renovasi/setuju/(:num)'] = 'beranda/setuju/$1';
$route['renovasi/tolak/(:num)'] = 'beranda/tolak/$1';
$route['renovasi/selesai/(:num)'] = 'beranda/doneRenovasi/$1';
$route['manage/user_baru'] = 'manage/createUser';
$route['manage/user_update/(:num)'] = 'manage/updateUser/$1';
$route['ajuan'] = 'beranda/tambahRenovasi';
$route['renovasi/ed/(:num)'] = 'beranda/ubahRenovasi/$1';
$route['gantipassword(:num)'] = 'ver_login/changepwd/$1';

// redirect direct access
$route['renovasi'] = 'beranda';

```

Gambar 5.2 Pengaturan *Routing* untuk Memanggil Fungsi

Tangkapan layar menunjukkan routing yang digunakan pada aplikasi web. Pada framework CI routing digunakan untuk memanggil fungsi-fungsi *controller* CI melalui uri web. Seiring dengan pengembangan aplikasi beserta fungsi-fungsi yang digunakan maka konfigurasi routing CI juga akan bertambah.

5.1.2. Halaman utama aplikasi

Controller utama yang digunakan untuk memproses halaman utama dashboard adalah Beranda.php dengan menggunakan Beranda_model.php untuk melakukan koneksi dengan database. View yang digunakan untuk menampilkan halaman utama terbagi menjadi 4, yaitu header.php, navigation.php, beranda_view.php, dan footer.php. Tampilan halaman utama

juga dibedakan menjadi 2 yaitu dalam keadaan non-login dan saat pengguna telah melakukan login. Pembagian tampilan dilakukan dengan membaca fungsi *session* pada framework CI dan view saat pengguna dalam keadaan login berada pada folder tersendiri. Untuk halaman utama view login yang digunakan berada pada `masuk/beranda_view.php`.

```

public function index()
{
    $data = $this->data;
    $data['luasTotal'] = $this->Beranda_model->totalLuas();
    if ($this->session->flashdata('cari')) {
        $data['listGedung'] = $this->session->flashdata('cari');
    } else {
        $data['invalid'] = $this->session->flashdata('invalid');
        if (isset($data['userLogin']) &&
            $data['userLogin']['userLevel']==4) {
            $priority = $this->Beranda_model->getListGedung('sarpras');
            if ($priority!=NULL) {
                $data['listGedung'] = $priority;
            } else {
                $data['noPriority'];
                $data['listGedung'] = $this->Beranda_model-
>getListGedung();
            }
        } else {
            $data['listGedung'] = $this->Beranda_model-
>getListGedung();
        }
    }
    $gd=0; foreach ($data['listGedung'] as $row) {
        if ($row['kategoriGedung']==1) {
            $data['listGedung'][$gd]['tipeGedung'] = 'university';
        } else {
            $data['listGedung'][$gd]['tipeGedung'] = 'building';
        }
        $gd++;
    }
    if (isset($this->data['userLogin'])) {
        if ($this->session->userdata('defaultPass')) {
            $data['modalPwd'] = $this->load-
>view('modal/modal_password', $data, TRUE);
        }
    }
    if ($data['userLogin']['userLevel']==4) {
        $this->load->view('template/header', $data);
        $this->load->view('template/navigation', $data);
        $this->load->view('template/menu', $data);
    }
}

```

Gambar 5.3 Fungsi Utama *Controller* Utama Beranda

Fungsi index sebagai fungsi utama pada Beranda adalah mengambil data gedung yang akan ditampilkan pada peta. Pada

awalnya controller akan mengambil semua data gedung yang tersedia, namun terdapat fungsi search untuk mengambil hanya data yang diinginkan.

```

function __construct()
{
    parent::__construct();
    $this->load->model('Beranda_model');
    $this->load->library('Encryption');
    $this->load->library('pagination');
    if ($this->uri->segment(1)=='beranda' || $this->uri->
    >segment(1)=='gedung' || $this->uri->uri_string()=='') {
        $this->session->unset_userdata('referred_from_renovasi');
        $this->session->unset_userdata('referred_from');
    }
    if ($this->session->userdata('logged_in')) {
        $this->data['userLogin'] = $this->session->
    >userdata('logged_in');
        switch ($this->session->
    >userdata['logged_in']['userLevel']) {
            case 1:
            case 2:
            case 5:
                if (is_numeric($this->uri->segment(2))) {
                    $this->data['jumlah'] = $this->Beranda_model->
    >jumlahRenovasi('ALL', (int)$this->uri->segment(2));
                } elseif ($this->uri->segment(2)=='pekerjaan' &&
    is_numeric($this->uri->segment(3))) {
                    $this->data['jumlah'] = NULL;
                } else {
                    $this->data['jumlah'] = $this->Beranda_model->
    >jumlahRenovasi('ALL', NULL, (int)$this->
    >data['userLogin']['userAuth']);
                }
                break;
            case 4:
                $this->data['jumlah'] = $this->Beranda_model->
    >jumlahRenovasi('ALL');
                $this->data['jumlahTersedia'] = $this->
    >Beranda_model->jumlahRenovasi('spr', NULL, (int)$this->
    >data['userLogin']['userAuth']);
                break;
            case 3:
                $this->data['jumlahBelum'] = $this->Beranda_model->
    >jumlahRenovasi('0');

```

Gambar 5.4 Fungsi Konstruktor Beranda

Pembagian tampilan non-login dan login pengguna diatur melalui pengecekan *session* pada fungsi `__construct` Beranda.php. Dalam keadaan login, *session* berisi username, nama lengkap pengguna dan level otentikasi pengguna. Fungsi konstruktor juga mengambil data – data dari database yang digunakan pada fungsi – fungsi isi sesuai dengan level otentikasi pengguna.

```

<?php
defined('BASEPATH') or exit('No direct script access
allowed');

class Beranda_model extends CI_Model
{
function __construct()
{
    parent::__construct();
}

/**
 * database read data gedung minimal
 * @method getListGedung
 * @param mixed $mode list gedung yang diambil
 * @return array sql result
 */
public function getListGedung($mode = 'full')
{
    $this->db->select('gedung.idGedung, kodeGedung,
namaGedung, luasGedung, x, y, kategoriGedung');
    $this->db->from('gedung');
    $this->db->join('koordinat', 'koordinat.idKoord =
gedung.koordGedung', 'left');
    $this->db->order_by('namaGedung', 'asc');
    $this->db->limit(8);
    if ($mode=='sarpras') {
        $this->db->join('proposal', 'proposal.idGedung =
gedung.idGedung', 'right');
        $this->db->join('pekerjaan', 'pekerjaan.idProposal =
proposal.idProposal', 'right');
        $this->db->where(array(
            'proposal.status' => '2',
            'pekerjaan.status' => '0',
            'gedung.pemilikGedung' => $this->session-
>userdata['logged_in']['userAuth']
        ));
        $this->db->group_by('idGedung');
    }
    $query=$this->db->get();
    if ($query->num_rows() > 0) {

```

Gambar 5.5 Fungsi-Fungsi Model Halaman Utama

Sebagai bagian dari pengembangan halaman utama juga dilakukan penulisan kode pada model `Beranda_model.php`. Fungsi utama yang dikembangkan pada model adalah untuk mengambil data kode, nama, dan luas gedung beserta lokasinya pada peta seluruh ITS.

Setelah itu dilakukan penulisan kode view untuk menampilkan data yang telah diambil ke tampilan web. Tampilan disajikan dalam bentuk dashboard gedung-gedung di ITS beserta lokasinya pada peta. Pengguna dapat berinteraksi dengan daftar gedung maupun ikon gedung pada peta untuk melihat data terkait gedung. Pengguna akan dialihkan ke halaman lain bila ingin melihat data gedung yang lebih lengkap.

5.1.3. Fungsi login

Fungsi login sendiri berada pada controller terpisah, yaitu pada `Ver_login.php`. Hal ini dilakukan dengan mempertimbangkan keamanan dan keandalan web sehingga tidak semua fungsi aplikasi ditangani oleh controller utama Beranda.

```

function __construct()
{
    parent::__construct();
    $this->load->model('Login_model');
}

/**
 * validate login
 * @method index
 * @return mixed user data
 * @todo change to use CI form validator
 */
function index()
{
    // Validation variable
    $username = $this->input->post('username');
    $password = $this->input->post('password');

    // Query db
    $result = $this->Login_model->login($username, $password);
    if ($result) {
        $sess_array = array();
        foreach ($result as $row) {
            $sess_array = array(
                'uid' => $row->uid,
                'username' => $row->username,
                'userLevel' => substr($row->user_level, 0, 1),
                'userAuth' => substr($row->user_level, 1),
                'namaLengkap' => $row->namaLengkap
            );

            $this->session->set_userdata( 'logged_in', $sess_array );
        }
        //check password default
        if ($password=='123qwe') {
            $this->session->set_userdata('defaultPass', 'changed');
        }
        redirect('beranda', 'refresh');
    } else {
        $this->session->set_flashdata('validUser', 'false');
        redirect('login', 'refresh');
    }
}

```

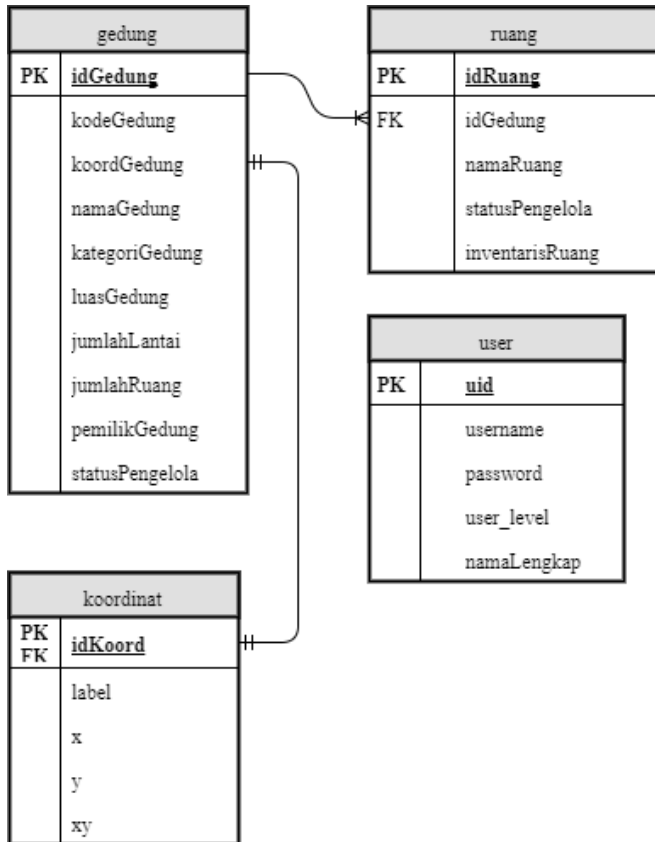
Gambar 5.6 Fungsi Login

Pada iterasi pertama ini evaluasi perangkat lunak dilakukan dengan metode *whitebox* karena perangkat lunak masih dalam tahap pengembangan *back-end* sehingga belum siap untuk dipresentasikan kepada pengguna. Evaluasi yang dilakukan oleh pengembang terkait fungsi CRUD perangkat lunak untuk mengambil dan mengubah data login yang dikembangkan sesuai *functional requirement*. Evaluasi aplikasi dijelaskan lebih lanjut pada bab VI mengenai hasil pengembangan aplikasi.

5.2. Purwarupa Kedua

Pada iterasi kedua pengembangan perangkat lunak dilakukan penulisan kode fungsi pengambilan data visual peta lahan dan gedung ITS baik secara keseluruhan maupun data individual per gedung. Pada tahap ini bagian *front-end* perangkat lunak mulai dikembangkan terkait dashboard utama dan *view* data gedung, namun tidak meliputi tampilan peta ITS karena data peta masih dikembangkan. Fungsi terkait berupa pengambilan data dari database dan tampilannya pada dashboard utama perangkat lunak.

Pada iterasi ini juga terdapat perubahan pada database berupa penambahan nama lengkap pengguna seperti tampak pada ERD berikut:



Gambar 5.7 Diagram Relasi Entitas Aplikasi Revisi Pertama

Perubahan lain yang dilakukan pada database adalah menghapus tabel kategori.

Kemudian pengembang melakukan *input* data awal pada database berdasarkan data yang sudah dimiliki oleh pihak SIMRI. Data awal yang dimasukkan ke database pada iterasi ini antara lain:

1. Nama gedung berdasarkan masterplan ITS tahun 2015,

2. Data ruang dan pengelola gedung masih dikosongi karena pihak SIMRI tidak menyediakan data terkait.

Kemudian pada iterasi ini pengembang juga melakukan pembangunan kode fungsi-fungsi data gedung yang hanya dapat dilakukan setelah pengguna *login* sesuai dengan perannya. Pengguna juga meminta adanya fungsi pencarian gedung untuk mempermudah melihat data gedung tanpa mencari satu persatu melalui tampilan peta. Fungsi yang dikembangkan dijelaskan sebagai berikut:

5.2.1. Fungsi data gedung

```
function detailGedung($ged)
{
    $data = $this->data;
    $result = $this->Beranda_model->getDataGedung($ged);
    if ($result) {
        $data['detailGedung'] = $result;
    } else {
        $data['detailGedung'] = NULL;
    }
    $data['renovasiLast'] = $this->Beranda_model->getListRenovasi($ged, 'last');
    $data['renovasiNow'] = $this->Beranda_model->getListRenovasi($ged, 'now');
    if ($data['renovasiLast'] != NULL) {
        $date = new
        DateTime($data['renovasiLast'][0]['dateDeleted']);
        $data['renovasiLast'][0]['dateDeleted'] = $date-
        >format('d-m-Y');
    }
    $array = array(
        'id' => $data['detailGedung'][0]['idGedung'],
        'url' => base_url().$this->uri->uri_string()
    );
    $this->session->set_userdata('referred_from', $array);
    $this->load->view('template/header', $data);
    $this->load->view('template/navigation', $data);
    if ($this->session->userdata('logged_in')) {
        $this->load->view('template/menu', $data);
    }
    $data['footer'] = $this->load->view('template/footer', NULL,
    TRUE);
    $this->load->view('gedung_view', $data);
}
```

Gambar 5.8 Fungsi Detail Data Gedung

Fungsi pada controller untuk menampilkan data gedung secara mendetail. Fungsi ini akan memanggil view baru yang berbeda dari halaman utama aplikasi.

```

public function getDataGedung(int $ged, $mode = NULL)
{
    if ($mode!=NULL) {
        $this->db->select('idGedung, kodeGedung, namaGedung,
luasGedung, tinggiGedung, jumlahLantai, kategoriGedung, x, y,
koordGedung');
        $this->db->join('koordinat', 'koordinat.idKoord =
gedung.koordGedung', 'left');
    } else {
        $this->db->select('idGedung, kodeGedung, namaGedung,
luasGedung, tinggiGedung, jumlahLantai, kategoriGedung');
    }
    $this->db->from('gedung');
    $this->db->where('idGedung', $ged);
    $query = $this->db->get();
    if ($query->num_rows() == 1) {
        return $query->result_array();
    } else {
        return null;
    }
}

```

Gambar 5.9 Fungsi Daftar Gedung

Fungsi pengambilan data dari database pada model CI. Data gedung yang diambil termasuk id gedung sebagai primary key, nama gedung, luas gedung, tinggi gedung, dan jumlah lantai yang dimiliki gedung.

5.2.2. Fungsi search

```
public function searchGedung()
{
    $data = $this->data;
    $this->load->library('form_validation');
    $this->form_validation->set_rules('cari_gedung', 'Search',
    'required');
    if ($this->form_validation->run() == false) {
        $this->session->set_flashdata('invalid', "Harap masukkan
    <strong>nama gedung</strong>");
    } else {
        $search = $this->input->post('cari_gedung');
        $result = $this->Beranda_model-
    >searchListGedungByName($search);
        if ($result) {
            $this->session->set_flashdata('cari', $result);
        } else {
            $this->session->set_flashdata('invalid', "Gedung
    tidak ditemukan. Apakah anda yakin <strong>nama
    gedung</strong> benar?");
        }
    }
    redirect('beranda');
}
```

Gambar 5.10 Fungsi Search Nama Gedung

Fungsi search bekerja dengan mencari gedung yang dicari ke model kemudian melakukan redirect dengan variabel sementara ke fungsi utama index sehingga data gedung terpopulasi dengan hasil pencarian. Bila gedung tidak ditemukan controller akan mengembalikan pengguna ke halaman utama dengan tampilan overview semua gedung dan memberi notifikasi bahwa nama gedung yang dicari tidak ditemukan. Fungsi juga akan menolak apabila pengguna tidak memasukkan karakter apapun ke dalam kotak search.


```

public function searchListGedungByName(string $ged)
{
    $this->db->select('idGedung, kodeGedung, namaGedung,
luasGedung, jumlahLantai, x, y, kategoriGedung');
    $this->db->from('gedung');
    $this->db->join('koordinat', 'koordinat.idKoord =
gedung.koordGedung', 'left');
    $this->db->where("MATCH(namaGedung) AGAINST ('$ged*' IN
BOOLEAN MODE)", null, false);
    $this->db->order_by('namaGedung', 'asc');

    $query=$this->db->get();
    if ($query->num_rows() > 0) {
        return $query->result_array();
    } else {
        return null;
    }
}

```

Gambar 5.11 Pengambilan Data Gedung dari Database

Fungsi pada model untuk mengambil data gedung sesuai parameter pencarian nama gedung. Fungsi ini memanfaatkan fitur *full-text index* dari database Mysql untuk membantu pencarian nama gedung yang panjang dan dapat melakukan pencarian kata secara parsial mengikuti sistem pencarian *natural language*.

Pada iterasi ini pengembang kembali belum melakukan evaluasi kepada pengguna karena belum tersedianya peta keseluruhan ITS sebagai dasar tampilan untuk ditunjukkan kepada pengguna.

5.3. Purwarupa Ketiga

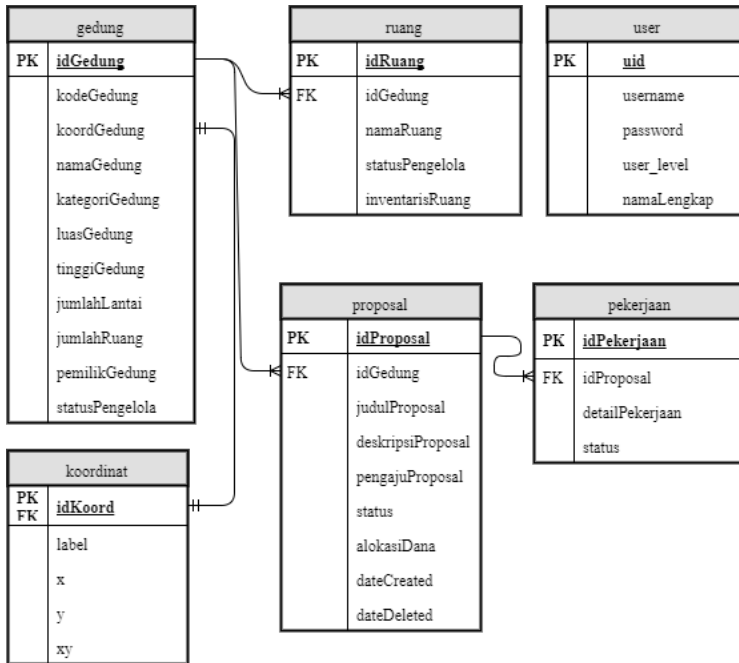
Iterasi ketiga pengembangan perangkat lunak membuat fungsi utama peninjauan dan perawatan per gedung dan statistik keadaan gedung secara menyeluruh dalam area ITS. Berdasarkan pada saran pengguna dari iterasi sebelumnya terdapat tambahan data gedung yang perlu ditunjukkan yaitu data ketinggian gedung sesuai data yang sudah ada.

Pengembangan *front-end* pada iterasi ini berupa memasukkan peta kekinian ITS yang diambil dari tugas akhir Rancang Bangun Peta Interaktif Tiga Dimensi Monitoring Lahan (MONLAH) di Institut Teknologi Sepuluh Nopember. Perlu diperhatikan bahwa peta yang diambil adalah peta dua dimensi yang menjadi dasar bentuk tiga dimensi yang menjadi hasil akhir dari penelitian tersebut. Untuk menampilkan peta dua dimensi yang telah tersedia pengembang menggunakan kode *javascript* yang telah tersedia yaitu *leaflet.js*. Terdapat fungsi tambahan yang dikembangkan pada iterasi ini yaitu fitur pengajuan renovasi gedung beserta pekerjaan renovasi terkait.

Fungsi peninjauan keadaan gedung berdasarkan pada status gedung yang telah disetujui oleh pihak SIMRI yaitu:

1. Gedung PENDIDIKAN, dimana yang termasuk adalah bangunan-bangunan ITS yang digunakan untuk keperluan pendidikan di ITS.
2. Gedung lain.

Dengan tambahan fitur-fitur yang diperlukan, maka database perangkat lunak juga diubah sehingga meliputi data ketinggian gedung dan data renovasi serta pekerjaan yang dilakukan. ERD database yang telah dirubah ditunjukkan pada gambar dibawah.



Gambar 5.12 Diagram Relasi Entitas Aplikasi Revisi Kedua

5.3.1. Fungsi renovasi gedung

Fungsi renovasi gedung mencakup data renovasi, pengajuan, persetujuan, dan perubahan ajuan renovasi. Pengembangan kode dimulai dari model untuk berkomunikasi dengan database.

```

function getListRenovasi(string $ged, $mode = NULL)
{
    if (!is_numeric($mode)) {
        switch ($mode) {
            case 'last':
                $this->db->select('idProposal, judulProposal,
dateDeleted');
                $this->db->where(array('idGedung' => $ged,
'dateDeleted !=' => NULL));
                $this->db->order_by('dateDeleted', 'desc');
                $this->db->limit(1);
                break;
            case 'now':
                $this->db->select('idProposal, judulProposal,
dateCreated');
                $this->db->where(array('dateDeleted' => NULL, 'idGedung'
=> $ged));
                $this->db->order_by('dateCreated', 'desc');
                $this->db->limit(1);
                break;
            default:
                $this->db->select('idProposal, judulProposal,
deskripsiProposal, status');
                $this->db->where('idProposal', $ged);
                $this->db->order_by('dateCreated', 'asc');
                break;
        }
    } else {
        $this->db->select('proposal.idProposal, gedung.idGedung,
namaGedung, judulProposal, deskripsiProposal,
gedung.pemilikGedung, proposal.status, alokasiDana,
dateCreated, dateDeleted');
        $this->db->select('(SELECT COUNT(*) FROM pekerjaan WHERE
pekerjaan.idProposal = proposal.idProposal AND
pekerjaan.status=1) as done', false);
        $this->db->select('(SELECT COUNT(*) FROM pekerjaan WHERE
pekerjaan.idProposal = proposal.idProposal) as kerja', false);
        $this->db->join('gedung', 'gedung.idGedung =
proposal.idGedung', 'right');
        $this->db->join('pekerjaan', 'pekerjaan.idProposal =
proposal.idProposal', 'left');
    }
}

```

Gambar 5.13 Fungsi Pengambilan Data Renovasi

Fungsi pengambilan data pada model memfasilitasi berbagai parameter sehingga dapat digunakan untuk mengambil data renovasi mengikuti keperluan aplikasi. Sebagai contoh untuk mengambil seluruh data gedung atau hanya gedung tertentu sesuai hak akses pengguna, atau dari penggunaan fitur *search*.

```
function createRenovasi($data)
{
    $sql = "INSERT INTO `proposal` (`idGedung`, `judulProposal`,
    `deskripsiProposal`, `pengajuProposal`, `dateCreated`) VALUES
    ('".$data['idGedung']."', '".$data['judulProposal']."',
    '".$data['deskripsiProposal']."',
    '".$data['pengajuProposal']."', CURDATE())";
    $this->db->query($sql);
    return $this->db->affected_rows();
}
```

Gambar 5.14 Fungsi *create* Renovasi

Fungsi untuk menambahkan data proposal yang baru diajukan ke dalam database. Data judul, deskripsi, dan pengaju proposal dikirim dari controller, sedangkan data tanggal pengajuan diambil dari tanggal data dimasukkan ke database.

```
function updateRenovasi(int $renovasi, array $data)
{
    $this->db->where('idProposal', $renovasi);
    $this->db->update('proposal', $data);
    return $this->db->affected_rows();
}
```

Gambar 5.15 Fungsi *update* Renovasi

Model untuk mengubah data renovasi pada database. Fungsi ini dipanggil oleh bermacam fungsi dari controller utama bukan hanya untuk melakukan perubahan data judul dan deskripsi renovasi. Fungsi juga dipanggil untuk melakukan perbaruan status renovasi pada fitur – fitur persetujuan, penolakan, dan menandai renovasi selesai.

```
function deleteRenovasi(int $renovasi)
{
    $this->db->delete('proposal', array('idProposal' =>
    $renovasi));
    return $this->db->affected_rows();
}
```

Gambar 5.16 Fungsi *delete* Renovasi

Fungsi untuk menghapus data renovasi pada database. Dalam implementasinya fungsi ini hanya dapat digunakan oleh pengguna admin dan pegawai SIMRI. Fungsi ini juga hanya digunakan pada saat maintenance untuk menyamakan data pada sistem dengan arsip yang dimiliki pihak pengelola gedung. Setelah model jadi, pengembangan dilanjutkan ke controller aplikasi.

```

function dataRenovasi($ged)
{
    $data = $this->data;
    switch ($ged) {
        case 'proposal':
            $result = $this->Beranda_model->getListRenovasi($ged, 2);
            break;
        case 'kerja':
            $result = $this->Beranda_model->getListRenovasi($ged, 3);
            break;
        case 'available':
            $result = $this->Beranda_model->getListRenovasi($ged, 4);
            break;
        default:
            $result = $this->Beranda_model->getListRenovasi($ged, 1);
            break;
    }

    if ($result!=null) {
        $data['dataRenovasi'] = $result;
        $d=0;
        foreach ($data['dataRenovasi'] as $row) {
            if ($row['kerja']!=0) {
                $data['dataRenovasi'][$d]['progress'] =
round(($row['done']/$row['kerja'])*100);
            }
            $date = new DateTime($row['dateCreated']);
            $data['dataRenovasi'][$d]['dateCreated'] = $date-
>format('d-m-Y');
            if (isset($row['dateDeleted']) &&
$row['dateDeleted']!=NULL) {
                $date = new DateTime($row['dateDeleted']);
                $data['dataRenovasi'][$d]['dateDeleted'] = $date-
>format('d-m-Y');
            }
            $d++;
        }
    }
    if ($ged!='ALL') {
        $renovasi = array(
            'id' => $data['dataRenovasi'][0]['idGedung'],
            'url' => base_url().$this->uri->uri_string()
        );
    }
}

```

Gambar 5.17 Fungsi untuk Menampilkan Daftar Renovasi

Fungsi pada controller menentukan user yang diperbolehkan untuk melakukan fungsi-fungsi pengajuan, perubahan, dan persetujuan renovasi yang ingin dilakukan. Fungsi ini juga menentukan tampilan *default* pengguna sesuai dengan hak akses pengguna pada saat login. Hak akses pengguna telah diterangkan pada bab IV mengenai perancangan aplikasi dan hak akses pengguna terdaftar.

5.3.2. Fungsi pekerjaan renovasi

Fungsi pekerjaan yang dilakukan pada renovasi terkait fungsi-fungsi CRUD database dengan pembagian sesuai dengan otentikasi pengguna. Model menunjukkan fungsi CRUD database dalam lingkup pekerjaan dijelaskan pada gambar berikut.

```
function getPekerjaan($kerja, $mode)
{
    $this->db->select('idPekerjaan, detailPekerjaan,
    pekerjaan.status, proposal.idProposal, judulProposal,
    deskripsiProposal, alokasiDana, dateCreated, dateDeleted');
    $this->db->from('proposal');
    $this->db->join('pekerjaan', 'pekerjaan.idProposal =
    proposal.idProposal', 'left');
    if ($mode==1) {
        $this->db->select('proposal.status as persetujuan,
        gedung.pemilikGedung AS unit');
        $this->db->join('gedung', 'gedung.idGedung =
        proposal.idGedung', 'left');
        $this->db->where('proposal.idProposal', $kerja);
    } elseif ($mode==2) {
        $this->db->where('pekerjaan.idPekerjaan', $kerja);
    }

    $query = $this->db->get();
    if ($query->num_rows()>0) {
        return $query->result_array();
    } else {
        return null;
    }
}
```

Gambar 5.18 Fungsi *read* Data Pekerjaan

Fungsi untuk mengambil data pekerjaan dari database. Fungsi model ini mengambil data pekerjaan berdasarkan parameter yang ditentukan dari controller. Fungsi juga digunakan untuk mengambil jumlah pekerjaan untuk dilakukan perhitungan progress renovasi pada controller.

```
function deletePekerjaan($kerja)
{
    $this->db->delete('pekerjaan', array('idPekerjaan' =>
    $kerja));
    return $this->db->affected_rows();
}
```

Gambar 5.19 Fungsi *delete* Pekerjaan

Fungsi untuk menghapus data pekerjaan. Fungsi ini dimaksudkan untuk menghapus pekerjaan yang menurut pengguna terkait tidak perlu dilakukan. Atau untuk menghapus pekerjaan yang sudah termasuk pada pekerjaan lain saat dilakukan perubahan pekerjaan.

```
function updatePekerjaan($data, $kerja = NULL)
{
    if ($kerja!=NULL) {
        $this->db->where('idPekerjaan', $kerja);
        $this->db->update('pekerjaan', $data);
        return $this->db->affected_rows();
    } else {
        return $this->db->update_batch('pekerjaan', $data,
        'idPekerjaan');
    }
}
```

Gambar 5.20 Fungsi *update* Pekerjaan

Fungsi untuk mengubah data pekerjaan. Fungsi mengambil data dari form perubahan pekerjaan yang berisi data pekerjaan sebelumnya. Fungsi ini juga digunakan pada fitur pengecekan pekerjaan untuk mengubah status pekerjaan.

```
function createPekerjaan($data)
{
    $this->db->insert('pekerjaan', $data);
    return $this->db->affected_rows();
}
```

Gambar 5.21 Fungsi *create* Pekerjaan

Fungsi untuk menambah pekerjaan baru. Data pekerjaan dikirim dari controller untuk dimasukkan ke database. Tiap melakukan penambahan data, status pekerjaan selalu dalam keadaan 'belum dikerjakan'.

```

function listPekerjaan($kerja)
{
    $data = $this->data;
    if ($this->session->flashdata('message')) {
        $data['message'] = $this->session->flashdata('message');
    }
    $data['back'] = $this->session->
    >userdata['referred_from_renovasi']['url'];
    $result = $this->Beranda_model->getPekerjaan((int)$kerja, 1);
    $data['dataPekerjaan'] = $result;

    $pekerjaan = array(
        'id' => $data['dataPekerjaan'][0]['idProposal'],
        'url' => base_url().$this->uri->uri_string()
    );
    $this->session->set_userdata('referred_from', $pekerjaan);

    if ($result!=null) {
        $d=0;
        foreach ($result as $row) {
            $date = new DateTime($row['dateCreated']);
            $data['dataPekerjaan'][$d]['dateCreated'] = $date->
            >format('d-m-Y');
            if ($row['dateDeleted']!=NULL) {
                $date = new DateTime($row['dateDeleted']);
                $data['dataPekerjaan'][$d]['dateDeleted'] = $date->
                >format('d-m-Y');
            }
            $data['idModal'] = $row['idPekerjaan'];
            $data['modal'][$row['idPekerjaan']] = $this->load-
            >view('modal/modal_delete', $data, TRUE);
            $d++;
        }
    }

    if ($data['userLogin']['userLevel']==4 &&
    $data['userLogin']['userAuth']!= $data['dataPekerjaan'][0]['uni
    t'] && $data['dataPekerjaan'][0]['idPekerjaan']!=NULL) {
        $data['warn'] = 'Anda tidak diperbolehkan mengisi ceklis
        pekerjaan pada renovasi gedung diluar unit Anda';
    }
}

```

Gambar 5.22 Fungsi-fungsi Pengambilan Data dan Pengubahan Data pada Database

Controller melakukan perubahan data sesuai hak akses pengguna melalui view yang ditentukan. Pada hak akses SARPRAS, controller akan mengubah daftar pekerjaan dalam bentuk *checklist* untuk mempermudah pengecekan. Pada hak akses pengguna lainnya, aplikasi akan menampilkan daftar pekerjaan berikut statusnya.

Setelah konstruksi kode selesai, pengembang membuat skenario pengujian pengguna dan melakukan evaluasi menggunakan *blackbox testing* kepada pengguna. Skenario *use case* untuk evaluasi pengguna dicantumkan pada bagian lampiran.

5.4. Purwarupa Keempat

Dalam implementasi pengembangan perangkat lunak, ternyata pengembangan dilakukan lebih dari tiga iterasi untuk mencakup perubahan yang diajukan oleh pengguna pada evaluasi iterasi sebelumnya. Pada iterasi keempat ini dilakukan pengembangan frontend terkait data pengguna, agar administrator aplikasi nantinya dapat menambah, mengubah, dan menghapus pengguna tanpa mengakses database secara langsung.

5.4.1. Fungsi data pengguna

Fungsi dikembangkan untuk mempermudah administrator perangkat lunak dalam mengelola pengguna yang dapat melakukan login, atau untuk membantu pengguna yang tidak dapat menggunakan sistem sebagaimana mestinya.

```

class Manage extends CI_Controller {

/**
 * load required model
 * @method __construct
 */
function __construct()
{
parent::__construct();
$this->load->model('Manage_model');
}

/**
 * tampilkan list user
 * @method index
 * @return array list user
 */
function index()
{
if ($this->session->userdata('logged_in')) {
    $data['all_user'] = $this->Manage_model->getUser();
    $u=0;
    foreach ($data['all_user'] as $row) {
        if ($data['all_user'][$u]['user_level']=='1'){
            $data['all_user'][$u]['userLevel']='Administrator';
        } elseif ($data['all_user'][$u]['user_level']=='2') {
            $data['all_user'][$u]['userLevel']='Pegawai SIMRI';
        } elseif ($data['all_user'][$u]['user_level']=='3') {
            $data['all_user'][$u]['userLevel']='Wakil Rektor
II';
        } elseif ($data['all_user'][$u]['user_level']=='4') {
            $data['all_user'][$u]['userLevel']='SARPRAS';
        } elseif ($data['all_user'][$u]['user_level']=='5') {
            $data['all_user'][$u]['userLevel']='Unit
Fakultas/Jurusan';
        } else {
            $data['all_user'][$u]['userLevel']='Pengguna Lain';
        }
        $data['idModal'] = $row['uid'];
        $data['modal'][$row['uid']] = $this->load-
>view('modal/modal_delete', $data, TRUE);

```

Gambar 5.23 Fungsi Tampilan Utama *Management User*

Aplikasi akan menunjukkan daftar pengguna yang ada pada database beserta hak akses masing-masing. Keseluruhan fungsi *management user* ditaruh pada controller tersendiri dengan pertimbangan hanya pengguna admin saja yang memiliki hak akses untuk menggunakan fitur ini.

```
function createUser()
{
    if ($this->session->userdata('logged_in')) {
        $data['userLogin'] = $this->session->userdata('logged_in');

        $this->load->library(array('form_validation'));
        $this->form_validation->set_rules('usernameForm', 'Username',
            'required');
        $this->form_validation->set_rules('passwordForm', 'Password',
            'required');
        $this->form_validation->set_rules('namaLengkapForm', 'Nama
            Lengkap', 'required');
        $this->form_validation->set_rules('user_levelForm', 'User
            Level', 'required');

        if ($this->form_validation->run() == FALSE)
        {
            $data['mode']="insert";
            $data['encPass'] = md5('test');
            $this->load->view('template/header', $data);
            $this->load->view('masuk/manage_form', $data);
        } else {
            $send = array(
                'username'=>$this->input->post('usernameForm'),
                'password'=>md5($this->input->post('passwordForm')),
                'namaLengkap'=>$this->input->post('namaLengkapForm'),
                'user_level'=>$this->input->post('user_levelForm')
            );
            $result = $this->Manage_model->createUser($send);
            if ($result==1) {
                $this->session->set_flashdata('message', 'Berhasil
membuat user baru');
            }
            redirect('manage');
        }
    }
}
```

Gambar 5.24 Fungsi *create User Baru*

Fungsi untuk menambahkan pengguna baru memanfaatkan fitur validasi yang disediakan CI. Pada form terdapat validasi untuk memastikan semua data yang diperlukan telah dimasukkan pengguna.

```
function deleteUser($id)
{
$result = $this->Manage_model->deleteUser($id);
if ($result==1) {
    $this->session->set_flashdata('message', 'User berhasil
dihapus');
} else {
    $this->session->set_flashdata('message', 'Gagal');
}
redirect('manage','refresh');
}
```

Gambar 5.25 Fungsi *delete* User

Fungsi untuk menghapus data pengguna. Fungsi ini akan menampilkan dialog warning sebelum dieksekusi dan memberikan notifikasi apabila data pengguna berhasil dihapus.

```

function updateUser($id)
{
    if ($this->session->userdata('logged_in')) {
        $data['userLogin'] = $this->session-
        >userdata('logged_in');
    }
    $this->load->library(array('form_validation', 'encrypt'));
    $this->form_validation->set_rules('usernameForm', 'Username',
    'required');
    $this->form_validation->set_rules('namaLengkapForm', 'Nama
    Lengkap', 'required');
    $this->form_validation->set_rules('user_levelForm', 'User
    Level', 'required');

    if ($this->form_validation->run() == FALSE)
    {
        $data['all_user'] = $this->Manage_model->editUser($id);
        $data['mode'] = "edit";
        $this->load->view('template/header', $data);
        $this->load->view('masuk/manage_form', $data);
    } else {
        $send = array(
            'username'=>$this->input->post('usernameForm'),
            'namaLengkap'=>$this->input->post('namaLengkapForm'),
            'user_level'=>$this->input->post('user_levelForm')
        );
        $result = $this->Manage_model->updateUser($id, $send);
        if ($result==1) {
            $this->session->set_flashdata('message', 'User
berhasil di update');
        }
        redirect('manage');
    }
}

```

Gambar 5.26 Fungsi *update* User

Fungsi untuk mengubah data pengguna oleh admin. Seperti fungsi untuk menambah pengguna, fungsi ini juga menggunakan validasi data yang dimasukkan pengguna. Sedangkan tangkapan layar kode pada model untuk berkomunikasi dengan database dapat dilihat pada gambar berikut.


```

function getUser()
{
$this->db->select('uid, username, user_level, namaLengkap');
$this->db->from('user');

$query = $this->db->get();
if ($query->num_rows()>0) {
    return $query->result_array();
} else {
    return null;
}
}

```

Gambar 5.27 Fungsi Utama pada Manage_model

Fungsi utama yang dieksekusi pada model akan mengambil data seluruh pengguna yang teregistrasi pada sistem. Seluruh data yang diambil ditampilkan dalam bentuk daftar pada laman daftar pengguna.

```

function createUser($send)
{
$this->db->insert('user', $send);
return $this->db->affected_rows();
}

```

Gambar 5.28 Fungsi-fungsi Penambahan Data User pada Database

Fungsi penambahan pengguna pada model. Fungsi akan memasukkan data pengguna baru yang didapat dari form pada controller ke dalam database pengguna.

```

/**
 * db ambil data user existing
 * @method editUser
 * @param int $id uid
 * @return array db rows
 */
function editUser($id)
{
    $this->db->select('uid, username, password, user_level,
namaLengkap');
    $this->db->from('user');
    $this->db->where('uid', $id);

    $query = $this->db->get();
    if ($query->num_rows()>0) {
        return $query->result_array();
    } else {
        return null;
    }
}

function updateUser($id, $send)
{
    $this->db->where('uid', $id);
    $this->db->update('user', $send);
    return $this->db->affected_rows();
}

function deleteUser($id)
{
    $this->db->delete('user', array('uid' => $id));
    return $this->db->affected_rows();
}

```

Gambar 5.29 Fungsi Perubahan dan Penghapusan Data User

Pada model diatas terdapat dua fungsi editUser dan updateUser, kedua fungsi tersebut memenuhi tujuan yang sama untuk mengubah data user yang sudah ada pada database. Juga ditampilkan fungsi untuk menghapus pengguna dengan parameter dari controller. Fungsi-fungsi pada model untuk melakukan perubahan data user juga ditaruh pada class tersendiri untuk menghindari disksekusinya kode yang tidak diperlukan pada model utama. Hal ini juga meringankan beban

sistem apabila pengguna admin tidak sedang menggunakan fitur manajemen pengguna.

5.5. Purwarupa Kelima

Saat melakukan uji coba dan presentasi pengguna pada iterasi keempat, ternyata pengguna mengajukan penambahan *functional requirement* sebagai fungsi tambahan untuk memfasilitasi fungsi utama yang telah ada pada perangkat lunak. Sehingga dalam implementasi pengembangan perangkat lunak dilakukan hingga iterasi kelima untuk mencakup penambahan FR tersebut. Pada iterasi ini, beberapa fungsi yang diminta oleh pengguna termasuk fungsi *CRUD* data gedung untuk mempermudah melakukan pembaruan data gedung tanpa membuka database secara langsung, fungsi untuk melakukan reset password pengguna, dan fungsi untuk melakukan upload data pekerjaan dalam bentuk excel.

5.5.1. Fungsi *CUD* data gedung

Pada iterasi – iterasi sebelumnya, aplikasi menggunakan data gedung permanen, dimana aplikasi hanya berfungsi untuk membaca (read) database gedung. Sehingga apabila terdapat perubahan data gedung dikarenakan perluasan, perubahan pengelola, dan sebagainya maka pengguna harus melakukan perubahan langsung pada database. Karena hal ini dianggap kurang praktis bagi pengguna, maka ditambahkan fungsi ini untuk mempermudah pengguna ‘admin’ maupun ‘SIMRI’ untuk melakukan perubahan dan penambahan data gedung sesuai keadaan terkini melalui antarmuka sistem, tanpa perlu melakukan pengubahan secara langsung melalui database. Perlu diperhatikan bahwa fungsi untuk menambahkan dan mengubah serta menghapus data gedung yang ada hanya untuk mempermudah pengelolaan dan perbaruan data gedung pada aplikasi, dan tidak mengikuti proses bisnis pihak pengelola gedung dalam melakukan pembangunan gedung baru, maupun merobohkan gedung lama.

```

function tambahGedung()
{
    $data = $this->data;
    $data['mode']="insert";
    $data['cancel'] = $this->session-
>userdata['referred_from']['url'];

    $this->load->library('form_validation');
    $this->form_validation->set_rules('namaGedungForm', 'nama
gedung', array(
        'required', 'callback__regex_check'
    ), array(
        'required' => 'Harap masukkan {field}'
    ));
    $this->form_validation->set_rules('luasGedungForm', 'luas
gedung', array(
        'decimal'
    ), array(
        'decimal' => 'Harap masukkan {field} dalam bentuk
desimal'
    ));
    $this->form_validation->set_rules('tinggiGedungForm', 'tinggi
gedung', array(
        'decimal'
    ), array(
        'decimal' => 'Harap masukkan {field} dalam bentuk
desimal'
    ));
    $this->form_validation->set_rules('jumlahLantaiForm', 'jumlah
lantai', array(
        'decimal'
    ), array(
        'decimal' => 'Harap masukkan {field} dalam bentuk
desimal'
    ));
    $this->form_validation->set_error_delimiters('<div
class="invalid-feedback">', '</div>');

    if ($this->form_validation->run() == false) {
        $this->load->view('template/header', $data);
        $this->load->view('template/navigation', $data);
    }
}

```

Gambar 5.30 Fungsi Menambah Data Gedung

Sesuai keperluan pengguna, aplikasi memberikan fitur untuk menambahkan data gedung baru. Fungsi ini menggunakan fitur validasi CI untuk mengecek data masukan pengguna sesuai dengan format database. Validasi yang dilakukan pada bagian luas gedung, tinggi gedung, dan jumlah lantai memastikan masukan berupa bilangan desimal.

5.5.2. Fungsi reset password pengguna

Fungsi dikembangkan untuk membantu pengguna apabila lupa akan password dan tidak dapat login ke aplikasi. Fungsi reset password hanya dapat dijalankan oleh pengguna dengan user level admin dan akan mengembalikan password pengguna ke password default yaitu '123qwe'.

```
function resetPassword($id)
{
    $result = $this->Manage_model->resetPwd($id);
    if ($result==1) {
        $this->session->set_flashdata('message',
        'Password user telah di reset ke password default "123qwe"');
    } else {
        $this->session->set_flashdata('message',
        'Gagal melakukan reset password');
    }
    redirect('manage','refresh');
}
```

Gambar 5.31 Fungsi Reset Password pada *controller* Manage

Fungsi untuk melakukan reset password pengguna pada controller. Fungsi akan menampilkan notifikasi saat password pengguna berhasil direset atau saat terjadi error.

```
function resetPwd($id)
{
    $this->db->where('uid', $id);
    $this->db->update('user', array(
        'password'=>md5('123qwe')
    ));
    return $this->db->affected_rows();
}
```

Gambar 5.32 Fungsi Reset Password pada *Manage_model*

Fungsi reset password pada model digunakan untuk mengubah data pengguna pada database. Setelah password di reset, pengguna akan mendapatkan notifikasi untuk melakukan pengubahan password tiap kali pengguna tersebut login selama password belum diubah.

```
//check password default
private $default_password = '123qwe';
if ($password==$default_password) {
    $this->session->set_userdata('defaultPass', 'changed');
}

//set warning for default password
if ($this->session->userdata('defaultPass')) {
    $data['modalPwd'] = $this->load-
>view('modal/modal_password', $data, TRUE);
}
```

Gambar 5.33 Fungsi Cek Password Default

Bersama dengan fitur untuk melakukan reset password ini maka juga diperlukan fitur bagi pengguna untuk melakukan penggantian password. Pengguna dapat melakukan penggantian password setelah melakukan login, dan pengguna dengan password default akan diingatkan untuk melakukan penggantian password untuk menghindari hal-hal yang tidak diinginkan. Fungsi pengecekan password ini diletakkan pada konstruktor fungsi utama agar selalu dieksekusi saat pengguna membuka laman utama aplikasi.

```

function changepwd($id)
{
    if (!$this->session->userdata('logged_in')) {
        $this->session->set_flashdata('warn', 'Anda belum
melakukan login!');
        redirect('beranda','refresh');
    }
    $data['userLogin'] = $this->session->userdata('logged_in');
    $data['cancel'] = base_url();
    $this->load->library('form_validation');
    $this->form_validation->set_rules('sandiLewat', 'password
lama', array(
        'callback__check_pass'
    ));
    $this->form_validation->set_rules('sandiLewatBaru', 'password
baru', array(
        'required', 'min_length[6]'
    ), array(
        'required' => 'Harap masukkan {field}',
        'min_length' => 'Password minimal 6 karakter'
    ));
    $this->form_validation->set_error_delimiters('<div
class="invalid-feedback">', '</div>');

    if ($this->form_validation->run() == FALSE) {
        $this->load->view('template/header', $data);
        $this->load->view('template/navigation', $data);
        $this->load->view('masuk/password_form', $data);
        $this->load->view('template/footer', $data);
    } else {
        $send = array(
            'password'=>md5($this->input->post('sandiLewatBaru'))
        );
        $result = $this->Login_model->chPwd((int)$id, $send);
        if ($result==1) {
            $this->session->set_flashdata('message', 'Password
anda berhasil diubah');
            $this->session->unset_userdata('pwd');
        } else {
            $this->session->set_flashdata('message', 'Perubahan
password gagal, mohon coba kembali');
        }
    }
}

```

Gambar 5.34 Fungsi Perubahan Password Pengguna

Untuk pengisian password sendiri terdapat beberapa batasan dimana password yang digunakan harus memiliki minimum 6

karakter alphanumeric dan tidak sama dengan username pengguna. Pengguna juga harus memasukkan password lama untuk dapat melakukan perubahan password.

5.5.3. Fungsi unggah excel pekerjaan

Fungsi upload excel ditujukan untuk mempermudah pengguna SIMRI dalam memasukkan data pekerjaan yang perlu dilakukan pada renovasi. Fungsi akan mengambil list pekerjaan dari file excel yang diunggah untuk diproses masuk ke dalam sistem. File excel yang diunggah harus memiliki kolom berjudul 'deskripsi pekerjaan' atau hanya memiliki kolom tunggal berisi daftar detil pekerjaan.


```

public function readExcel()
{
    $data = $this->data;

    $config['upload_path'] = './files/excel/';
    $config['allowed_types'] = 'xls|xlsx|csv';
    $config['max_size'] = 10000;

    $this->upload->initialize($config);

    if (! $this->upload->do_upload('excelFileForm')) {
        $error = $this->upload->display_errors();
        $this->session->set_flashdata('message', $error);
    } else {
        $data['upload_data'] = $this->upload->data('file_name');
        $new_name = $this->upload-
>data('file_path').'.uid'.$data['userLogin']['uid'].'_('.time()
        .')_'. $this->upload->data('orig_name');
        rename(
            $this->upload->data('full_path'),
            $new_name
        );

        $inputFileName = $new_name;
        try {
            $inputFileType =
IOFactory::identify($inputFileName);
            $reader = IOFactory::createReader($inputFileType);
            $spreadsheet = $reader->load($inputFileName);
        } catch (Exception $e) {
            unlink($new_name);
            die('Error loading file "'. $data['upload_data']['.':
'.'. $e->getMessage());
        }

        $sheet = $spreadsheet->getSheet(0);
        $highestRow = $sheet->getHighestRow();
        $highestColumn = $sheet->getHighestColumn();
        $colNumber =
Coordinate::columnIndexFromString($highestColumn);

```

Gambar 5.35 Fungsi Baca File Excel

Fungsi ini akan membaca file excel yang diunggah oleh pengguna. Pada fungsi ini dilakukan validasi file yang diunggah pengguna. Validasi yang dilakukan terkait jenis file yang diunggah dan format daftar pekerjaan dalam file excel tersebut.

```
function importPekerjaan(array $data)
{
    $result = $this->db->insert_batch('pekerjaan', $data);
    return $result;
}
```

Gambar 5.36 Fungsi Model untuk Memasukkan Pekerjaan

Kode pada controller akan membaca daftar detail pekerjaan baris per baris untuk kemudian diinputkan ke dalam database. Pengguna akan dikembalikan ke laman detail pekerjaan dengan notifikasi apabila proses unggah daftar pekerjaan dalam bentuk excel berhasil atau bila proses gagal beserta alasannya. Aplikasi akan menampilkan error apabila format daftar pekerjaan tidak sesuai dengan format yang diperlukan aplikasi.

Halaman ini sengaja dikosongkan

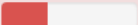





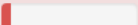
BAB VI

HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai hasil pengembangan aplikasi pada tiap tahap iterasi. Hasil yang dijelaskan berikut merupakan rangkuman hasil evaluasi pengguna setelah purwarupa aplikasi pada tiap tahap dipresentasikan kepada pengguna.

6.1. Hasil dan Evaluasi Purwarupa Pertama

Pada iterasi pertama ini pengembang belum melakukan evaluasi kepada pengguna, melainkan melakukan testing internal terhadap purwarupa 1 untuk memastikan *functional requirement* yang telah dikembangkan bekerja sebagaimana mestinya dari sudut pandang *backend* aplikasi. Sehingga pada tahap ini pengembang menggunakan unit testing menggunakan PHPunit untuk memastikan bahwa fungsi pada controller dan model telah berjalan sebagaimana mestinya. Hasil testing dicantumkan sebagaimana berikut:

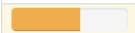


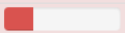
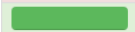
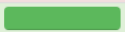
		Lines	
Total		33.97%	106 / 312
 controllers		76.19%	64 / 84
 models		100.00%	27 / 27
 views		7.46%	15 / 201

Legend

Low: 0% to 50% Medium: 50% to 90% High: 90% to 100%

Gambar 6.1 *Code coverage* Hasil Unit Testing Purwarupa Pertama

Hasil unit testing diatas menunjukkan code coverage atau method PHP yang telah di cek menggunakan PHPunit. Kolom 'Lines' menunjukkan cakupan baris kode yang dijalankan dan memberikan hasil yang telah didefinisikan. Kolom sebelah kiri menunjukkan cakupan coverage pada bagian-bagian MVC kode, pada bagian controller, model, dan view.

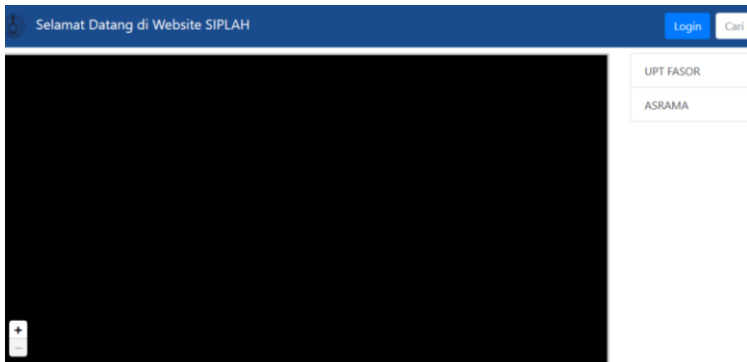
Code Coverage					
Functions and Methods			Classes and Traits		
	58.82%	10 / 17		50.00%	3 / 6
	46.15%	6 / 13		25.00%	1 / 4
	100.00%	4 / 4		100.00%	2 / 2
	n/a	0 / 0		n/a	0 / 0

Gambar 6.2 Lanjutan *Code coverage* Hasil Unit Testing Purwarupa Pertama

Kolom ‘Functions and Methods’ melakukan tes yang sama dalam cakupan fungsi pada kode PHP, dan kolom ‘Classes and Traits’ menunjukkan hasil eksekusi kode dalam cakupan tiap kelas PHP. Disini baris view memiliki coverage paling sedikit pada kolom lines karena menggunakan framework CI, meskipun view berupa file php namun tidak mengandung fungsi logika aplikasi dan kelas terdefinisi. Pada view hanya terdapat fungsi sederhana untuk menampilkan variabel-variabel php yang dikirim dari controller. Dapat dilihat pada bagian view hanya kolom ‘Lines’ yang tersedia karena mengandung beberapa baris PHP tersebut.

6.2. Hasil dan Evaluasi Purwarupa Kedua

Pada iterasi kedua tampilan frontend aplikasi telah mulai dikembangkan. Namun perlu diingat seperti telah dijelaskan pada bab sebelumnya bahwa pada iterasi kedua ini peta terkini seluruh ITS masih dalam tahap pembuatan sehingga aplikasi belum dapat menampilkan peta tersebut.



Gambar 6.3 Tampilan Utama Aplikasi pada Purwarupa Pertama

Tangkapan layar diatas menunjukkan tampilan dashboard utama aplikasi yang menunjukkan data gedung di ITS secara keseluruhan. Tampilan ini tersedia untuk semua pengguna tanpa melakukan login terlebih dahulu.



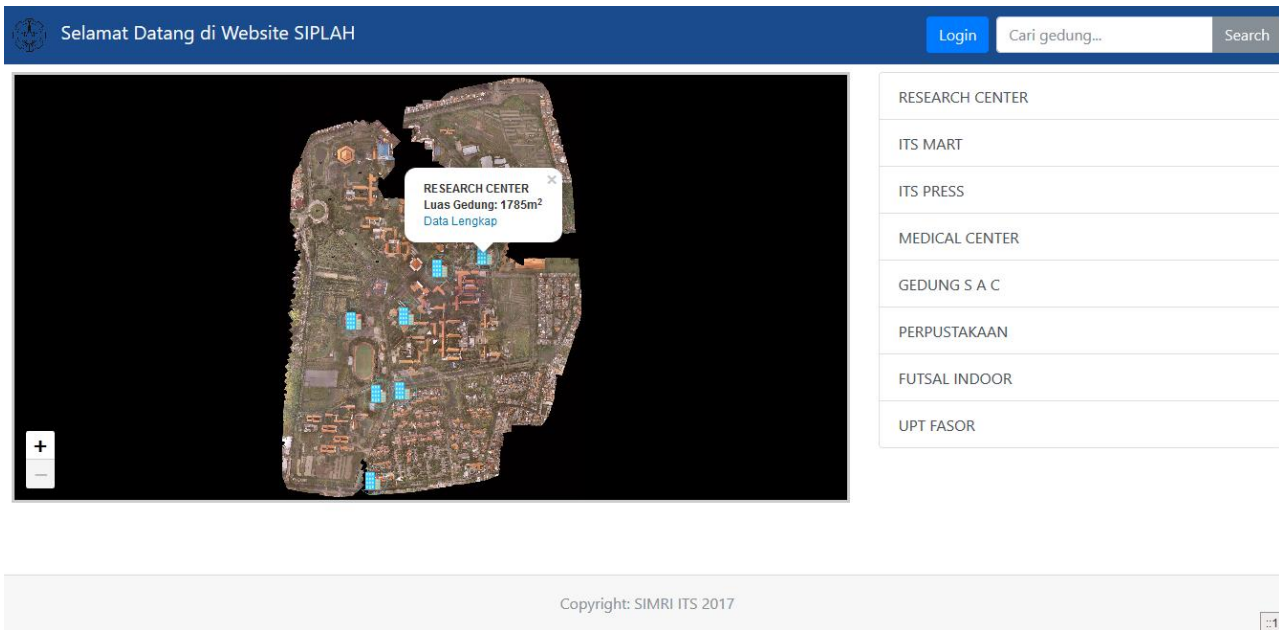
Gambar 6.4 Tampilan Utama Aplikasi Saat Pengguna Sudah Login

Tangkapan layar menunjukkan aplikasi pada saat pengguna telah melakukan login. Terdapat perbedaan bagi pengguna yang telah login yaitu terdapat menu untuk melihat detail data gedung, renovasi, dan pekerjaan sesuai dengan otentikasi pengguna.

Pada gambar diatas pengembang melakukan login dengan otentikasi admin (superuser) yang dapat mengubah seluruh data pada sistem, sehingga semua fungsi tersedia. Pada gambar 6.3 dan 6.4 terlihat bahwa aplikasi sudah memiliki fungsi untuk menampilkan peta, namun karena data belum diisi maka hanya nampak kontrol peta kosong. Pada kedua gambar tersebut juga sudah dimasukkan 2 data *dummy* gedung untuk pengujian aplikasi.

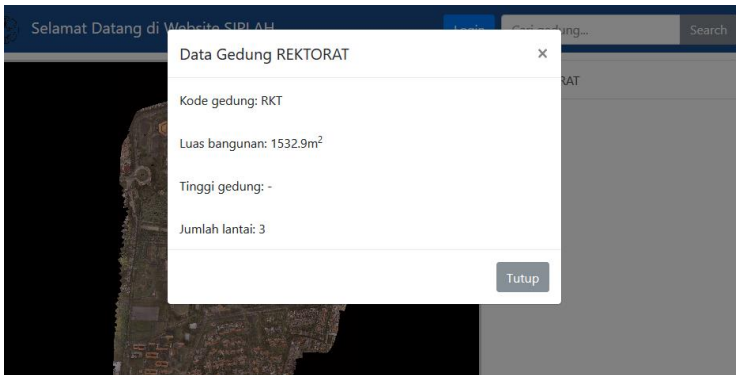
6.3. Hasil dan Evaluasi Purwarupa Ketiga

Pada iterasi ketiga luaran purwarupa sudah memiliki frontend yang lengkap dan dapat dipresentasikan kepada pengguna.



Gambar 6.5 Tampilan Akhir Halaman Utama Aplikasi

Tangkapan layar menunjukkan halaman utama aplikasi yang dapat diakses pengguna tanpa melakukan login. Pada tampilan ini fungsi yang dapat diakses pengguna terbatas pada login, search, dan data gedung beserta lokasinya pada peta ITS. Pengguna dapat melakukan klik pada daftar gedung atau ikon gedung pada peta untuk melihat detail data gedung.



Gambar 6.6 Tampilan Data Lengkap Gedung

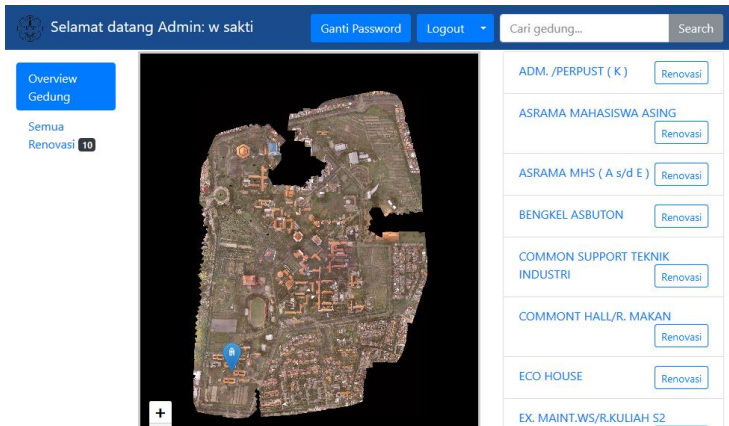
Tangkapan layar untuk halaman detail data gedung. Data yang ditampilkan sesuai dengan keperluan pengguna mencakup nama gedung, kode gedung, luas bangunan, tinggi gedung, dan jumlah lantai pada gedung. Tangkapan layar ini diambil saat pengguna login menggunakan hak akses ‘Pegawai SIMRI’.

Masukkan Login

Gambar 6.7 Tampilan Login User

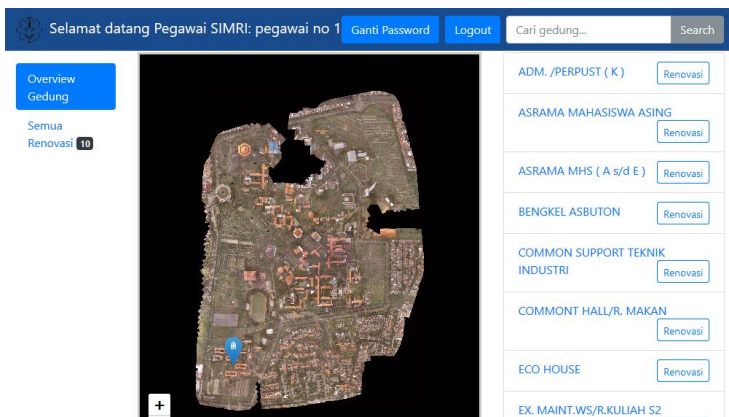
Tampilan form login pengguna. Untuk mengakses fitur lain yang membutuhkan hak akses, pengguna harus melakukan login terlebih dahulu. Dibawah ini akan dijelaskan mengenai

perbedaan tampilan halaman utama dan halaman pendukung lain untuk hak akses pengguna yang berbeda.



Gambar 6.8 Tampilan Login User Admin

Tangkapan layar diatas merupakan tampilan user ‘Admin’. Semua fungsi tersedia bagi user admin, mulai dari membuat dan mengubah data renovasi gedung, data pekerjaan renovasi, dan pengubahan data pengguna lain.



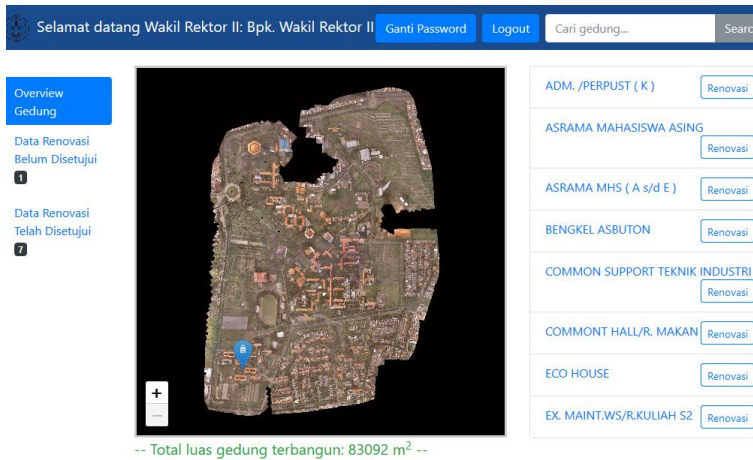
Gambar 6.9 Tampilan Utama User Pegawai SIMRI

Tangkapan layar halaman utama user dengan hak akses ‘Pegawai SIMRI’. Fitur yang tersedia mencakup pengubahan data renovasi dan menentukan data pekerjaan renovasi.



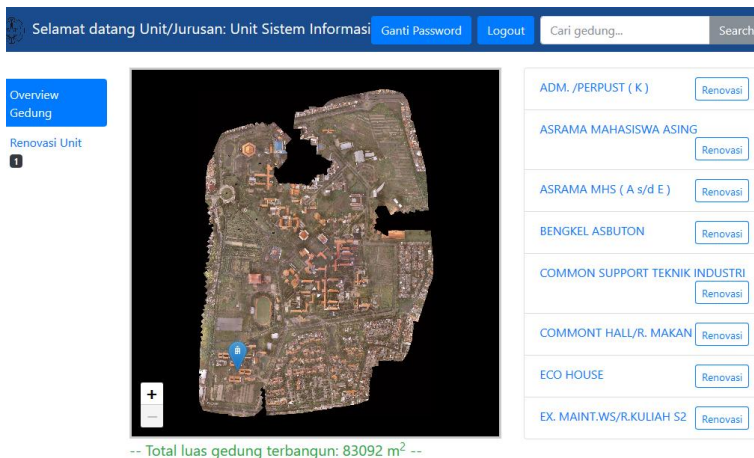
Gambar 6.10 Tampilan Utama User SARPRAS

Tangkapan layar halaman utama user SARPRAS. Berbeda dengan hak akses pengguna lain, pada halaman utama user SARPRAS hanya memunculkan data gedung yang memiliki data renovasi aktif, dengan data pekerjaan yang masih belum dilakukan. Pengguna SARPRAS masih dapat melihat data gedung lain, namun harus menggunakan fitur search.



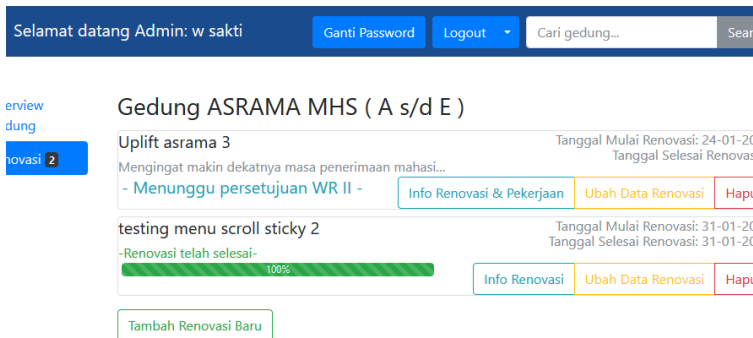
Gambar 6.11 Tampilan Utama User WR II

Halaman utama user WR II hampir sama dengan tampilan pengguna lainnya, dengan menunjukkan keseluruhan data gedung yang ada di ITS. Perbedaannya terletak pada menu navigasi, dimana user WR II hanya dapat melihat proposal yang belum disetujui atau proposal yang sudah disetujui.



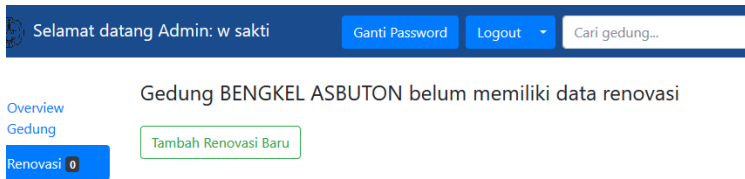
Gambar 6.12 Tampilan Utama User Unit Fakultas/Jurusan

Tampilan halaman utama user ‘Unit Fakultas/Jurusan’ kurang lebih sama dengan pengguna dengan hak akses ‘Pegawai SIMRI’. Hanya saja pengguna tidak dapat melakukan penambahan dan perubahan pekerjaan pada saat membuka tampilan data renovasi gedung.



Gambar 6.13 Tampilan Daftar Renovasi Gedung

Tampilan data renovasi gedung menunjukkan daftar renovasi yang sedang dan telah dilakukan pada gedung yang dipilih. Disini data yang ditampilkan merupakan data *dummy* karena pihak SIMRI belum memiliki data renovasi gedung yang siap untuk digunakan dalam aplikasi. Perlu diperhatikan tangkapan layar diambil dengan menggunakan login user ‘Pegawai SIMRI’ sehingga semua fungsi terkait renovasi dapat digunakan. Untuk pengguna lain seperti ‘Unit’ hanya dapat mengajukan renovasi baru dan melihat data pekerjaan renovasi. Untuk pengguna ‘WR II’ terdapat fungsi untuk melakukan persetujuan renovasi dan mengisikan anggaran dana renovasi.



Selamat datang Admin: w sakti

Ganti Password Logout Cari gedung...

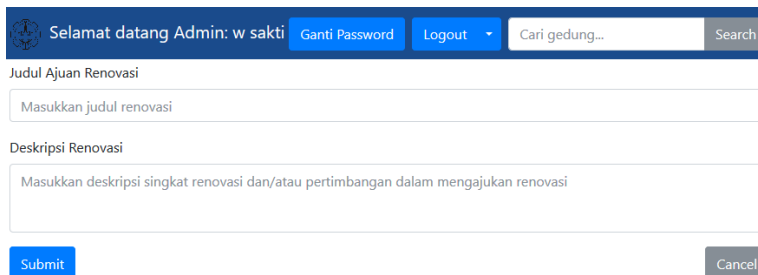
Overview
Gedung
Renovasi 0

Gedung BENGKEL ASBUTON belum memiliki data renovasi

Tambah Renovasi Baru

Gambar 6.14 Tampilan Gedung Tanpa Renovasi

Gedung yang tidak memiliki data renovasi akan menampilkan data kosong dan tombol ajuan renovasi.



Selamat datang Admin: w sakti Ganti Password Logout Cari gedung... Search

Judul Ajuan Renovasi

Masukkan judul renovasi

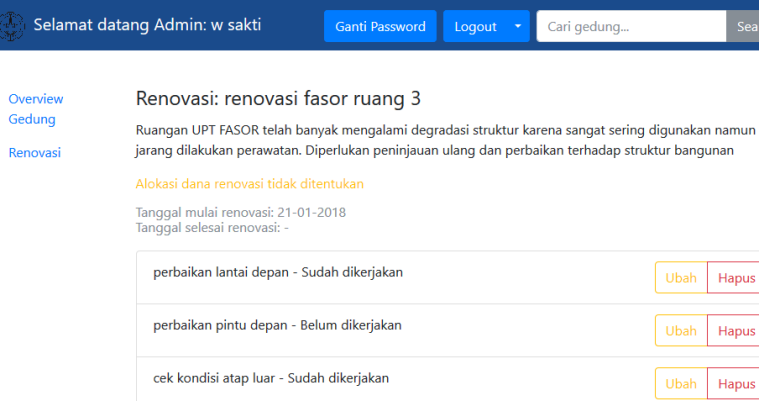
Deskripsi Renovasi

Masukkan deskripsi singkat renovasi dan/atau pertimbangan dalam mengajukan renovasi

Submit Cancel

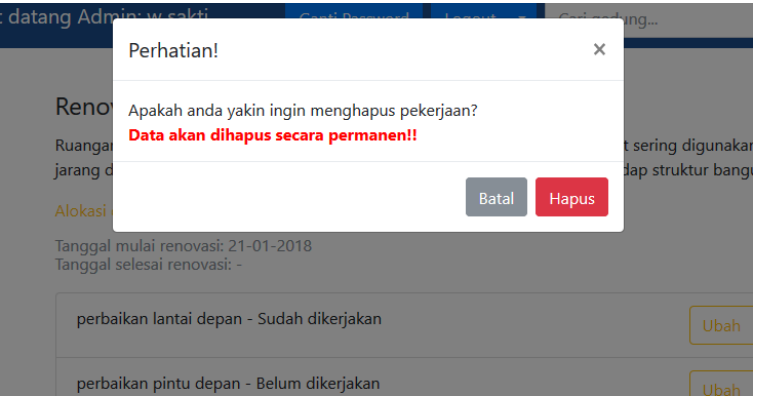
Gambar 6.15 Tampilan Pengajuan Proposal Renovasi

Tampilan untuk pengguna dengan hak akses untuk mengajukan proposal renovasi baru. Pengguna mengisi judul singkat dan mengutarakan pertimbangan-pertimbangan mengapa renovasi gedung perlu dilakukan.



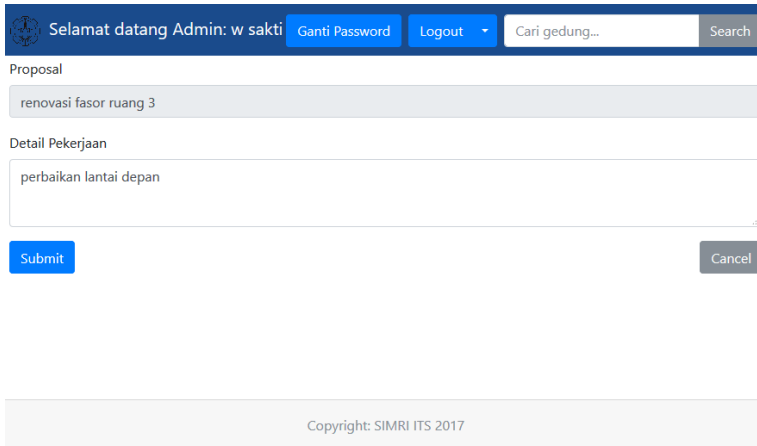
Gambar 6.16 Tampilan Detail Renovasi Termasuk Pekerjaan dan Proses Renovasi

Data pekerjaan renovasi menampilkan data-data renovasi seperti judul, deskripsi pertimbangan renovasi dengan lengkap, dan data pekerjaan renovasi yang sudah dan belum dilakukan, beserta proses renovasi berdasarkan data pekerjaan. Tangkapan layar user ‘Pegawai SIMRI’ dengan hak akses untuk menambahkan dan mengubah pekerjaan, pengguna lain hanya dapat melihat proses renovasi dan data pekerjaan namun tidak dapat melakukan perubahan lain.



Gambar 6.17 Tampilan Hapus Pekerjaan Renovasi

Saat pengguna akan menghapus renovasi aplikasi akan menampilkan kotak dialog warning untuk memastikan.



The screenshot displays the SIMRI ITS 2017 web application interface. At the top, a blue header bar contains the text 'Selamat datang Admin: w sakti' next to a profile icon, followed by 'Ganti Password' and 'Logout' buttons. A search bar with the placeholder 'Cari gedung...' and a 'Search' button is also present. Below the header, the 'Proposal' section shows 'renovasi fasor ruang 3'. The 'Detail Pekerjaan' section contains a text input field with the value 'perbaikan lantai depan'. At the bottom of this section are 'Submit' and 'Cancel' buttons. A footer bar at the very bottom states 'Copyright: SIMRI ITS 2017'.

Gambar 6.18 Tampilan Mengubah Data Pekerjaan

Presentasi dilakukan dengan menggunakan skenario yang telah dibuat kepada pihak SIMRI. Perlu disebutkan pengembang hanya melakukan validasi dan use case testing kepada pihak SIMRI dengan pertimbangan bahwa pihak sebagai pemangku kepentingan utama. Sehingga apabila pihak SIMRI telah memberikan persetujuan maka dapat dianggap bahwa proses pengembangan purwarupa aplikasi telah selesai dan sesuai dengan harapan pengguna. Pengembang juga meminta validasi dan feedback dari pengguna untuk dasar iterasi selanjutnya. Feedback yang diberikan oleh pengguna diterangkan sebagai berikut:

6.4. Hasil dan Evaluasi Purwarupa Keempat

Purwarupa keempat mencakup tambahan dan perubahan-perubahan sesuai permintaan pengguna pada iterasi-iterasi sebelumnya.

Username	Nama Lengkap	User Level	Action
admin	w sakti	Administrator	<div>Update UserReset PasswordDelete User</div>
pegawai1	pegawai no 1	Pegawai SIMRI	<div>Update UserReset PasswordDelete User</div>
wr2	Bpk. Wakil Rektor II	Wakil Rektor II	<div>Update UserReset PasswordDelete User</div>
sarpras1	SARPRAS unit FTI	Pengguna Lain	<div>Update UserReset PasswordDelete User</div>
pegawai2	pegawai simri 2	Pegawai SIMRI	<div>Update UserReset PasswordDelete User</div>
sisfor	Unit Sistem Informasi	Pengguna Lain	<div>Update UserReset PasswordDelete User</div>
elektro	test	Unit Fakultas/Jurusan	<div>Update UserReset PasswordDelete User</div>

Gambar 6.19 Tampilan *Management User*

Tangkapan layar diatas menunjukkan perubahan yang diminta oleh pengguna yaitu tampilan untuk mengelola data pengguna. Aplikasi menunjukkan daftar pengguna beserta hak akses masing-masing. Fitur ini hanya dapat diakses oleh user admin.

ID:

2

Username:

pegawai1

Nama Lengkap:

pegawai no 1

User Level:

2

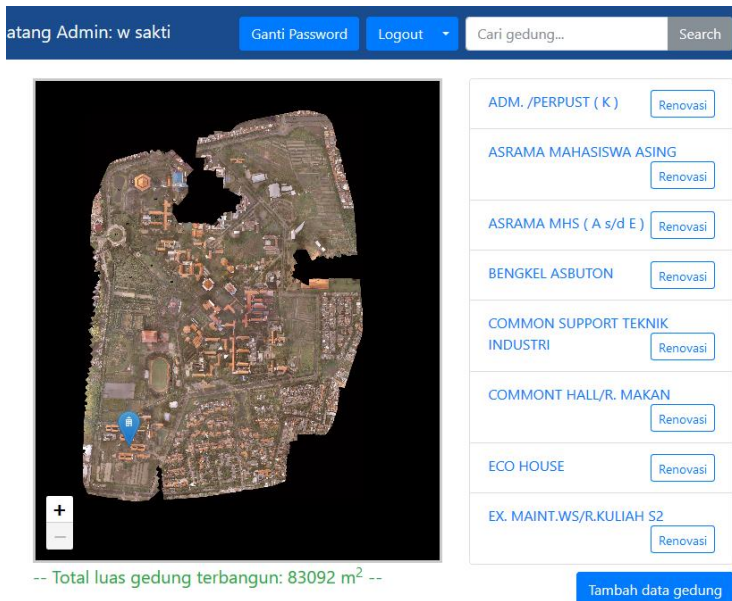
Submit

Gambar 6.20 Tampilan Ubah Data Pengguna

Tangkapan layar menunjukkan fitur untuk mengubah data pengguna. Laman ini diakses melalui opsi ‘update user’ pada daftar pengguna seperti terlihat pada gambar 6.19.

6.5. Hasil dan Evaluasi Purwarupa Kelima

Purwarupa kelima menambahkan fitur-fitur yang diminta pada iterasi sebelumnya yaitu fitur perubahan data gedung, reset password pengguna, dan fitur unggah excel untuk pekerjaan renovasi. Luaran dari purwarupa ini dianggap sudah final dan merupakan hasil akhir dari serangkaian iterasi pengembangan perangkat lunak.



Gambar 6.21 Tombol 'Tambah data gedung' pada Laman Utama

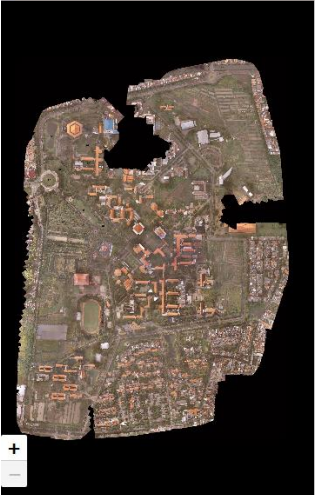
Fitur untuk menambahkan data gedung baru dapat diakses dari laman utama aplikasi. Di akhir daftar gedung terdapat tombol untuk menambahkan data gedung baru.

Selamat datang Admin: w sakti

Ganti Password

Logout

Cari gedung...



+

-

Nama Gedung

Masukkan nama gedung

Kode Gedung

Masukkan kode gedung (bila ada)

☐ Gedung pendidikan

Luas Bangunan

Masukkan luas bangunan

Tinggi Bangunan

Masukkan tinggi bangunan

Jumlah Lantai

Masukkan jumlah lantai gedung

Lokasi Gedung


Untuk menambah lokasi gedung lakukan klik pada peta.

Submit

Reset

Gambar 6.22 Laman Form Penambahan Gedung Baru

Untuk menambahkan data gedung baru pengembang memberikan link tambahan pada laman utama aplikasi. Form untuk menambahkan data gedung memiliki beberapa validasi isian dan tampilan peta untuk memilih lokasi gedung.

 Selamat datang Admin: w sakti

[Ganti Password](#) [Logout](#)

[Data Gedung](#)
[Renovasi](#) 2

Gedung ASRAMA MAHASISWA ASING

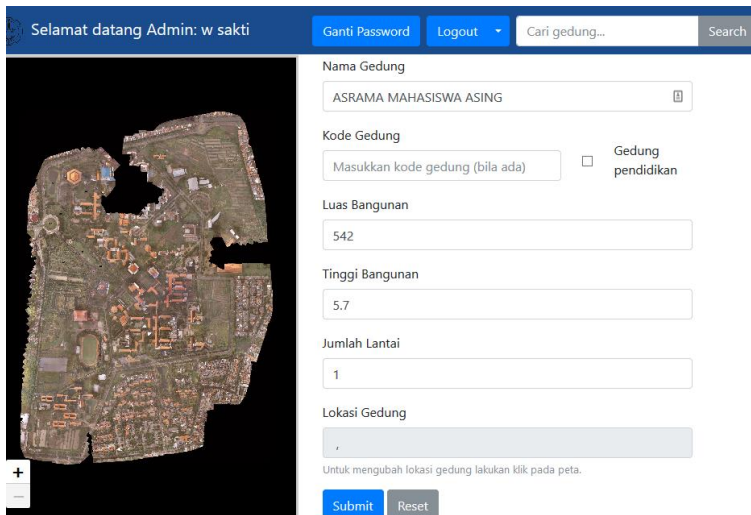
Kode gedung: - tidak ada -
Luas bangunan: 542m ²
Tinggi gedung: 5.7m
Jumlah lantai: 1
Renovasi sedang berjalan: test newline

[Ubah Data Gedung](#)

[Data Renovasi Lengkap](#)

Gambar 6.23 Tombol 'Ubah Data Gedung' pada Laman Detail Gedung

Untuk mengakses fitur perubahan data gedung pengguna perlu membuka laman detil data gedung terlebih dahulu. Pada laman ini terdapat opsi untuk mengubah data gedung. Perlu diperhatikan bahwa tata letak tombol dan data berubah mengikuti ukuran jendela browser pengguna.



Selamat datang Admin: w sakti

Ganti Password Logout Cari gedung... Search

Nama Gedung
ASRAMA MAHASISWA ASING

Kode Gedung
Masukkan kode gedung (bila ada) ☐ Gedung pendidikan

Luas Bangunan
542

Tinggi Bangunan
5.7

Jumlah Lantai
1

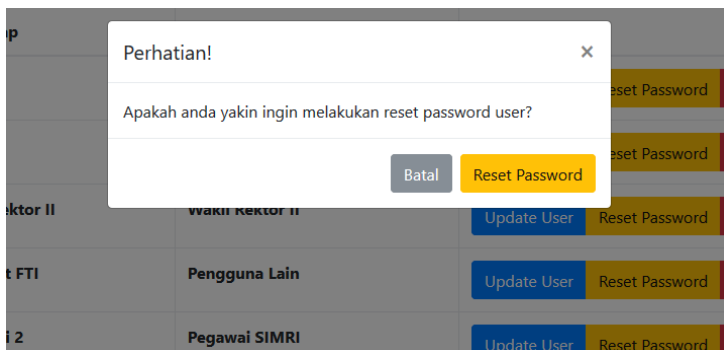
Lokasi Gedung
[Map Area]

Untuk mengubah lokasi gedung lakukan klik pada peta.

Submit Reset

Gambar 6.24 Laman Form Mengubah Data Gedung

Form yang ditampilkan menunjukkan data gedung pada database sekarang untuk dirubah sesuai keperluan pengguna.



Perhatian!

Apakah anda yakin ingin melakukan reset password user?

Batal Reset Password

No	Nama Pengguna	Aksi
1	Rektor II	Update User Reset Password
2	Pengguna Lain	Update User Reset Password
3	Pegawai SIMRI	Update User Reset Password

Gambar 6.25 Fungsi Reset Password Pengguna

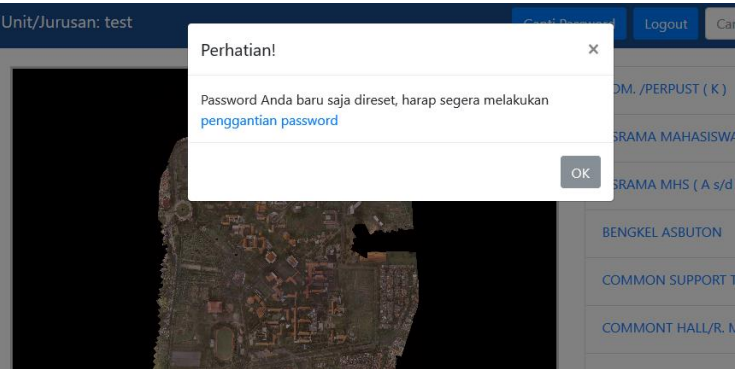
Fungsi reset password dapat diakses pengguna dengan hak akses admin melalui laman daftar pengguna. Aplikasi akan menampilkan pop-up warning untuk memastikan pengguna tidak salah.

Password user telah di reset ke password default "123qwe"

#	Username	Nama Lengkap	User Level	Action
1	admin	w sakti	Administrator	Update User
2	pegawai1	pegawai no 1	Pegawai SIMRI	Update User
3	wr2	Bpk. Wakil Rektor II	Wakil Rektor II	Update User
4	sarpras1	SARPRAS unit FTI	Pengguna Lain	Update User

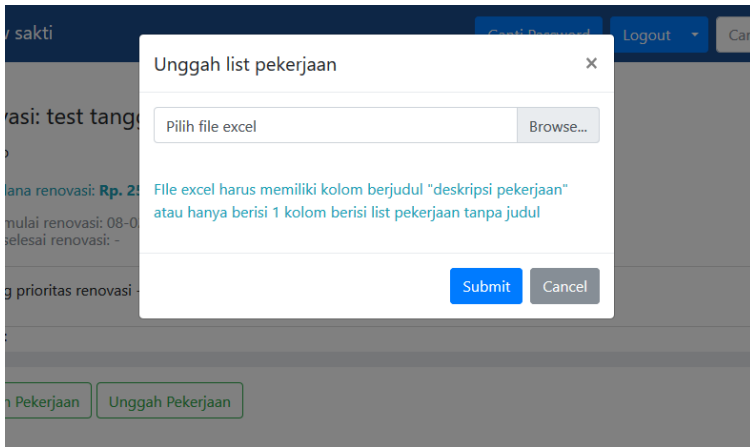
Gambar 6.26 Notifikasi Password Reset

Aplikasi juga akan menampilkan notifikasi apabila password telah berhasil direset beserta password default yang digunakan.



Gambar 6.27 Notifikasi pada Pengguna dengan Password Default

Saat pengguna yang telah direset passwordnya melakukan login, selama pengguna tersebut belum melakukan penggantian password maka pengguna akan diingatkan untuk mengganti passwordnya.



Gambar 6.28 Fungsi Unggah Excel Pekerjaan

Pengguna dapat mengunggah file excel berisi daftar pekerjaan untuk mempermudah pengisian pekerjaan renovasi. File yang diunggah berupa file excel dengan ekstensi .xls, .xlsx, atau .ods dan harus memiliki kolom berjudul "deskripsi pekerjaan" atau hanya berisi 1 kolom berisi list pekerjaan tanpa judul, seperti tertera pada kotak dialog pengunggahan. Aplikasi akan memunculkan error apabila ketentuan tersebut tidak dipenuhi atau terdapat error lain.

BAB VII

KESIMPULAN DAN SARAN

7.1. Kesimpulan

Setelah melalui seluruh proses pengerjaan tugas akhir dengan judul ‘Pengembangan Sistem Informasi Monitoring Pengelolaan Gedung di ITS’, maka dapat disimpulkan bahwa pengembangan aplikasi SIMPG telah dapat menyelesaikan rumusan masalah awal mengenai sistem informasi monitoring untuk kebutuhan pihak manajemen ITS. Hal ini terbukti dari proses pengujian aplikasi yang telah dilakukan. Selama proses pengerjaan tugas akhir ini juga didapatkan beberapa temuan dalam mengembangkan perangkat lunak, diantaranya penambahan beberapa FR untuk mempermudah pengguna dalam mengoperasikan aplikasi.

7.2. Saran

Pada akhir pengembangan aplikasi masih terdapat beberapa hal yang dapat diperbaiki dan dikembangkan lebih agar dapat membantu pihak SIMRI dalam melakukan peninjauan gedung di ITS. Beberapa hal yang dianggap masih kurang dan dapat ditingkatkan antara lain utamanya data gedung yang ada di ITS. Pada pengembangan aplikasi kali ini data yang didapat hanya data gedung tahun 2016, dimana data tersebut berisi data gedung yang sudah dibangun pada periode pembangunan sebelumnya yaitu di akhir tahun 2015. Dari data tersebut, data bangunan yang lengkap hanya terdapat pada luas bangunan. Sedangkan data ketinggian gedung tidak tersedia pada semua gedung. Sedangkan data gedung yang sedang dan baru selesai dibangun hingga saat penelitian ini selesai tidak disediakan pihak pengelola. Demikian juga dengan data ruang yang ternyata masih belum dimiliki oleh pihak SIMRI.

Dari segi aplikasi yang dikembangkan juga terdapat beberapa fitur yang masih dapat ditingkatkan seperti melakukan pengunggahan foto gedung yang ingin direnovasi dan proses pengerjaan renovasinya. Selain itu pada aplikasi masih belum

terdapat pembagian antar pengguna dengan otentikasi ‘Pegawai SIMRI’ dan ‘Pegawai unit fakultas/jurusan’. Seharusnya terdapat pembagian antar unit fakultas/jurusan yang berbeda untuk meningkatkan keandalan dan keamanan aplikasi sehingga tiap pengguna pada tingkat unit hanya dapat mengakses pengajuan renovasi pada unit masing-masing. Demikian juga pada tingkat pengguna SIMRI, seharusnya terdapat perbedaan antara pegawai dengan kepala bagian, dengan pembagian mengikuti deskripsi pekerjaannya. Namun hal ini memerlukan persetujuan dengan semua pihak termasuk pihak unit fakultas/jurusan sehingga masih belum dapat diwujudkan pada penelitian ini.

Pengembangan lebih lanjut juga dapat meliputi proses bisnis pembangunan gedung baru dan bukan hanya renovasi gedung yang sudah dibangun saja.

DAFTAR PUSTAKA

- [1] “Institut Teknologi Sepuluh Nopember.” [Daring]. Tersedia pada: <https://www.its.ac.id/article/profil-its/id>. [Diakses: 30-Sep-2016].
- [2] Institut Teknologi Sepuluh Nopember, “Peraturan Rektor Institut Teknologi Sepuluh Nopember Nomor 10 Tahun 2016,” 2016.
- [3] H. Son, F. Bosché, dan C. Kim, “As-built data acquisition and its use in production monitoring and automated layout of civil infrastructure: A survey,” *Adv. Eng. Informatics*, vol. 29, no. 2, hal. 172–183, 2015.
- [4] M. Hajdukiewicz, D. Byrne, M. M. Keane, dan J. Goggins, “Real-time monitoring framework to investigate the environmental and structural performance of buildings,” *Build. Environ.*, vol. 86, hal. 1–16, 2015.
- [5] M. P. Martin, “Prototyping,” in *Encyclopedia of Information Systems*, Elsevier, 2003, hal. 565–573.
- [6] T. Gartzzen, F. Brambring, dan F. Basse, “Target-oriented Prototyping in Highly Iterative Product Development,” *Procedia CIRP*, vol. 51, hal. 19–23, Jan 2016.
- [7] T. Ostrand, “Black-Box Testing,” in *Encyclopedia of Software Engineering*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2002.
- [8] S. Nidhra, ... J. D.-J. of E. S. and A., dan undefined 2012, “Black box and white box testing techniques-a literature review,” *academia.edu*.
- [9] T. Amanatidis dan A. Chatzigeorgiou, “Studying the evolution of PHP web applications,” *Inf. Softw.*

Technol., vol. 72, hal. 48–67, 2016.

- [10] “CodeIgniter at a Glance — CodeIgniter 3.1.7 documentation,” 24-Jan-2018. [Daring]. Tersedia pada: https://codeigniter.com/user_guide/overview/at_a_glance.html.
- [11] “Composer (software),” *Wikipedia*. 24-Jan-2018.
- [12] “PHPUnit Manual – Chapter 2. Writing Tests for PHPUnit,” 24-Jan-2018. [Daring]. Tersedia pada: <https://phpunit.de/manual/7.0/en/writing-tests-for-phpunit.html#writing-tests-for-phpunit.data-providers>.
- [13] kenjis, “ci-phpunit-test: An easier way to use PHPUnit with CodeIgniter 3.x.” 24-Jan-2018.
- [14] L. Slusky, “Integrating software modelling and prototyping tools,” *Inf. Softw. Technol.*, vol. 29, no. 7, hal. 379–387, 1987.
- [15] P. Shoval dan N. Pliskin, “Structured prototyping: Integrating prototyping into structured system development,” *Inf. Manag.*, vol. 14, no. 1, hal. 19–30, Jan 1988.

BIODATA PENULIS



Penulis bernama Theo Wibismana Sakti lahir di Malang Jawa Timur pada tanggal 1 Oktober 1993. Penulis merupakan anak pertama dari dua bersaudara. Penulis telah menempuh pendidikan formal di SDN Percobaan 1, SMP Negeri 1 Malang, dan SMA Negeri 1 Malang. Dan sekarang penulis sedang menuntut ilmu di Institut Teknologi Sepuluh Nopember pada Departemen Sistem Informasi

Fakultas Teknologi Informasi dan Komunikasi. Penulis masuk ITS menjadi angkatan 2011 melalui jalur tes atau SNMPTN dengan nomor induk (NRP) 5211100065. Selama masa perkuliahan, penulis aktif menjadi asisten praktikum pada bermacam mata kuliah. Selama dua periode penulis juga menjabat sebagai asisten laboratorium. Dan di akhir masa perkuliahan, penulis memilih topik Tugas Akhir pada bidang minat lab IKTI (Infrastruktur Keamanan Teknologi Informasi). Jika terdapat pertanyaan mengenai Tugas Akhir ini, penulis dapat dihubungi melalui e-mail theo.w.sakti@gmail.com atau nomor telepon +6283834784488.

Halaman ini sengaja dikosongkan

LAMPIRAN A

SKENARIO EVALUASI PERANGKAT LUNAK

Berikut terlampir skenario *black box testing* yang digunakan oleh pengembang pada saat melakukan evaluasi dengan pengguna. Evaluasi dengan pengguna baru dilakukan pada iterasi ketiga, keempat, dan kelima. Sedangkan pengujian purwarupa pertama dan kedua dilakukan menggunakan unit testing PHP, sehingga pada iterasi pertama dan kedua pengembang hanya mencantumkan daftar use case yang dilakukan pada saat unit testing.

A.2

Evaluasi Sistem Informasi Monitoring Pengelolaan Gedung di ITS (SIMPG)
(evaluasi dilakukan oleh pengguna – blackbox testing)

Skenario Evaluasi Perangkat Lunak

Iterasi Ketiga

Requirement skenario:

- Pengguna telah membuka website sistem monitoring
- Pengguna sudah memiliki login pada sistem

Fitur	Hasil/ Tanggapan Pengguna	
	Status	Keterangan/ Saran
Pemantauan gedung <ul style="list-style-type: none"> - (pengguna dapat melakukan zoom dan panning pada peta untuk mencari gedung) - Klik gedung yang ingin dilihat - Lihat data-data gedung 	✓	<ul style="list-style-type: none"> - Data gedung masih belum lengkap. - Data luas keseluruhan gedung
Search gedung <ul style="list-style-type: none"> - Pengguna melakukan pencarian menggunakan nama gedung pada kotak search - Klik pada peta atau pada daftar gedung 	✓	
Login <ul style="list-style-type: none"> - Klik login - Masukkan username dan password pengguna - Klik masuk 	✓	
Melihat data proses renovasi Req.: pengguna telah melakukan login <ul style="list-style-type: none"> - Cari dan/atau pilih gedung yang ingin dilihat renovasinya - Klik tombol 'renovasi' 	✓	
Pengajuan renovasi Req.: pengguna telah login dan memiliki hak akses untuk mengajukan renovasi <ul style="list-style-type: none"> - Pilih gedung yang ingin dilakukan renovasi - Klik 'Tambah renovasi' - Masukkan judul renovasi dan pertimbangan mengapa renovasi diperlukan - Klik 'Submit' 	✓	
Persetujuan renovasi Req.: pengguna telah melakukan login dengan hak akses untuk menyetujui renovasi <ul style="list-style-type: none"> - Klik pada daftar proposal renovasi 	✓	

Dibuat oleh: Theo W Sakti – 5211100065
Sistem Informasi – Institut Teknologi Sepuluh Nopember
21 Desember 2017

Evaluasi Sistem Informasi Monitoring Pengelolaan Gedung di ITS (SIMPG)
(evaluasi dilakukan oleh pengguna – blackbox testing)

<ul style="list-style-type: none"> - Masukkan anggaran dana renovasi - Klik 'Setujui renovasi' 		
<p>Penolakan renovasi Req.: pengguna telah melakukan login dengan hak akses untuk menolak renovasi</p> <ul style="list-style-type: none"> - Klik pada daftar proposal renovasi untuk menolak - Masukkan alasan penolakan (opsional) - Klik 'Tolak renovasi' 	✓	Proghess renovasi belum dimunculkan pada hkn renovasi
<p>Perubahan data renovasi Req.: pengguna telah login dengan hak akses untuk merubah data renovasi</p> <ul style="list-style-type: none"> - Cari gedung atau renovasi yang akan dirubah - Klik tombol 'Ubah' pada daftar renovasi - Lakukan perubahan data - Klik 'Submit' 	✓	
<p>Renovasi selesai Req.: Pengguna telah login dengan hak akses persetujuan selesai renovasi</p> <ul style="list-style-type: none"> - Cari dan pilih gedung atau renovasi - Klik tombol 'Renovasi selesai' 	✓	Bedakan view antara renovasi - sudah selesai
<p>Penambahan pekerjaan renovasi Req.: Pengguna telah login dengan hak akses menambahkan pekerjaan Pengguna sudah membuka halaman renovasi</p> <ul style="list-style-type: none"> - Klik tombol 'Info renovasi & pekerjaan' - Klik tombol 'Tambah pekerjaan' - Masukkan data pekerjaan yang perlu dilakukan pada renovasi - Klik 'Submit' 	✓	
<p>Perubahan pekerjaan renovasi Req.: Pengguna telah login dengan hak akses merubah pekerjaan Pengguna telah membuka halaman daftar pekerjaan</p> <ul style="list-style-type: none"> - Klik tombol 'Ubah' pada baris pekerjaan yang akan dirubah - Ubah data pekerjaan - Klik 'Submit' 	✓	

A.4

Evaluasi Sistem Informasi Monitoring Pengelolaan Gedung di ITS (SIMPG)
(evaluasi dilakukan oleh pengguna – blackbox testing)

Menghapus pekerjaan renovasi Req.: Pengguna telah login dengan akses menghapus pekerjaan renovasi Pengguna sudah membuka daftar pekerjaan - Klik tombol 'Hapus'	✓	
--	---	--

Evaluasi Sistem Informasi Monitoring Pengelolaan Gedung di ITS (SIMP)
(evaluasi dilakukan oleh pengguna – blackbox testing)

Skenario Evaluasi Perangkat Lunak

Iterasi Keempat

Requirement skenario:

- Pengguna telah membuka website sistem monitoring
- Pengguna telah melakukan login sebagai 'admin'

Fitur	Hasil/ Tanggapan Pengguna	
	Status	Keterangan/ Saran
Mengubah data pengguna <ul style="list-style-type: none"> - Buka laman daftar pengguna - Klik 'ubah' pada pengguna - Lakukan perubahan data - Klik 'Simpan' 	✓	
Menambah pengguna baru <ul style="list-style-type: none"> - Buka laman daftar pengguna - Klik 'Tambah' - Masukkan data pengguna - Klik 'Simpan' 	✓	
Menghapus pengguna <ul style="list-style-type: none"> - Buka laman daftar pengguna - Klik tombol 'Hapus' pada pengguna - Klik 'Ya' 	✓	

A.6

Evaluasi Sistem Informasi Monitoring Pengelolaan Gedung di ITS (SIMPG)
(evaluasi dilakukan oleh pengguna – blackbox testing)

Skenario Evaluasi Perangkat Lunak

Iterasi Kelima

Requirement skenario:

- Pengguna telah membuka website sistem monitoring
- Pengguna memiliki login sebagai admin atau SIMRI

Fitur	Hasil/ Tanggapan Pengguna	
	Status	Keterangan/ Saran
Mengubah data gedung Req.: pengguna telah melakukan login <ul style="list-style-type: none"> - (pengguna dapat melakukan zoom dan panning pada peta untuk mencari gedung) - Klik pada gedung yang ingin dirubah datanya - Klik 'Ubah data gedung' - Lakukan perubahan data - Klik 'Simpan' 	✓	
Menambahkan data gedung <ul style="list-style-type: none"> - Klik 'tambah data gedung' pada laman utama - Masukkan data gedung dan pilih lokasi pada peta - Klik 'simpan' 	✓	
Menghapus gedung <ul style="list-style-type: none"> - Klik pada gedung yang ingin dihapus - Klik 'Hapus gedung' - Klik 'Hapus' 	✓	
Melakukan reset password pengguna Req.: pengguna login sebagai 'admin' <ul style="list-style-type: none"> - Buka laman daftar pengguna - Klik 'reset password' - Klik 'Ya' 	✓	Tunjukkan password default biar admin tidak lupa.
Mengunggah file excel detail pekerjaan Req.: pengguna berada pada laman detail renovasi <ul style="list-style-type: none"> - Klik 'unggah pekerjaan' - Pilih file excel (pastikan format file excel benar) - Klik unggah - Cek pada laman detail renovasi apakah daftar pekerjaan sudah benar 	✓	Format file excel ditunjukkan waktu mengunggah file
Mengubah kata sandi pengguna <ul style="list-style-type: none"> - Klik 'ubah password' 	✓	

Dibuat oleh: Theo W Sakti – 5211100065

Sistem Informasi – Institut Teknologi Sepuluh Nopember

21 April 2017

Evaluasi Sistem Informasi Monitoring Pengelolaan Gedung di ITS (SIMPg)
(evaluasi dilakukan oleh pengguna – blackbox testing)

- Masukkan password lama dan password baru		
- Klik 'simpan'		

Halaman ini sengaja dikosongkan

LAMPIRAN B

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

Berikut terlampir dokumen spesifikasi kebutuhan perangkat lunak yang digunakan dalam pembangunan perangkat lunak pada tugas akhir ini.

B.2

Rekam Revisi

Tanggal	Versi	Deskripsi	Penulis
31-November-2017	1.0	Draft awal SRS	Theo W Sakti
16-September-2017	2.0	Perubahan kebutuhan fungsional	Theo W Sakti
03-Januari-2018	3.0	Penambahan kebutuhan fungsional mengikuti umpan balik pengguna	Theo W Sakti

Spesifikasi Kebutuhan Perangkat Lunak

1. Pengantar

Dokumen Software Requirement Specification (SRS – Spesifikasi Kebutuhan Perangkat Lunak) ini menjelaskan dengan detail kebutuhan-kebutuhan (requirements) aplikasi yang berhubungan dengan tugas akhir Pengembangan Sistem Informasi Monitoring Gedung di ITS. Dalam SRS ini akan dicantumkan deskripsi dan kebutuhan perangkat lunak baik itu merupakan perangkat lunak, perangkat keras, fungsi, komunikasi, dan penggunaannya.

1.1 Tujuan

Tujuan ditulisnya dokumen SRS ini adalah sebagai dasar pengembangan perangkat lunak dengan judul Sistem Monitoring Pengelolaan Gedung Institut Teknologi Sepuluh Nopember (SIMPG – ITS) sehingga luaran yang dihasilkan, yaitu produk berupa aplikasi berbasis web dapat memenuhi kebutuhan pengguna secara fungsional maupun non-fungsional seperti yang telah disetujui antara pengembang dan pengguna. Persetujuan kebutuhan perangkat lunak juga akan dijelaskan pada dokumen ini.

Mengenai perangkat lunak yang dikembangkan, fungsi dan tujuannya adalah agar pihak Subdirektorat Implementasi Rencana Induk (SIMRI) sebagai pengelola gedung di ITS mendapat manfaat perangkat lunak dalam melakukan pengelolaan gedung. Pengelolaan gedung yang dibantu oleh perangkat lunak khususnya berfokus pada perawatan/renovasi gedung di ITS.

1.2 Ruang Lingkup

Ruang lingkup dokumen SRS ini adalah mengenai deskripsi kebutuhan fungsional dan non-fungsional dari aplikasi yang akan dibuat, diantaranya adalah: keandalan (reliability), keamanan (security) aplikasi dan kinerja (performance) aplikasi berbasis web.

Sistem informasi yang dibuat akan mampu menampilkan data gedung di ITS secara tepat dan terkini, sehingga akan membantu pihak SIMRI dalam melakukan pemantauan kondisi gedung.

B.4

Sistem informasi juga dapat menampilkan luas keseluruhan bangunan pada lahan ITS, beserta lokasinya pada peta yang juga disediakan oleh aplikasi untuk mempermudah visualisasi dari posisi gedung. Sistem informasi yang dikembangkan juga didesain untuk menampilkan renovasi gedung yang sudah selesai, renovasi yang sedang dikerjakan, maupun untuk melakukan pengajuan dan persetujuan renovasi gedung.

1.3 Definisi, Akronim, dan Singkatan

1. PHP

PHP merupakan salah satu bahasa pemrograman yang dapat digunakan dalam membangun sebuah aplikasi berbasis web. Bahasa pemrograman PHP sangat sering digunakan pada berbagai aplikasi web di dunia dan merupakan salah satu dari 10 bahasa pemrograman yang dianggap paling matang. Penggunaan bahasa pemrograman PHP dalam penelitian ini mempertimbangkan implementasi dan operasi aplikasi yang mudah dan tidak menyusahkan pengguna. Dengan mengembangkan aplikasi berbasis web, pengguna akan lebih mudah untuk mengakses dan menggunakan aplikasi kapanpun dan dimanapun tersedia koneksi internet. Aplikasi juga mudah diakses baik menggunakan *browser* pada komputer ataupun menggunakan perangkat bergerak.

2. Codeigniter

CodeIgniter merupakan *framework* aplikasi PHP untuk membuat bahasa pemrograman PHP mengikuti sistem *Object Oriented Programming* menggunakan *Model-View-Controller* (MVC). *Framework* ini termasuk sangat populer dan sering digunakan oleh pengembang web pada umumnya. Penggunaan CI dalam melakukan pengembangan aplikasi berbasis web ditujukan untuk mempercepat dan mempermudah proses penulisan kode, karena CI menyediakan beragam *library* untuk

melakukan hal-hal yang biasa dilakukan menggunakan pemrograman PHP.

3. Composer

Composer adalah *package manager* atau pengelola paket dan dependensi pada level aplikasi web untuk bahasa pemrograman PHP. Composer memberikan standar untuk mengelola dependensi dan *library* yang diperlukan oleh suatu aplikasi PHP. Dalam konteks penelitian ini, composer digunakan untuk mengatur *library* PHPUnit dan ci-phpunit-test untuk pengujian aplikasi, dan ekstensi PHP lain seperti xDebug.

1.4 Referensi

1.4.1 Pengembangan Perangkat Lunak Prototyping

Prototyping Software Development Life Cycle (SDLC – Siklus Hidup Pengembangan Perangkat Lunak) adalah sebuah metodologi dalam melakukan pengembangan perangkat lunak dengan menciptakan prototype/purwarupa perangkat lunak dalam beberapa iterasi pengembangan hingga perangkat lunak memenuhi spesifikasi perangkat lunak yang telah disetujui antara pengembang dan stakeholder. Pengembangan perangkat lunak dengan metode prototyping menyediakan kerangka kerja pengembangan yang lebih fleksibel dan mudah dirubah dibandingkan dengan menggunakan metode pengembangan yang lebih tradisional seperti waterfall, spiral, dan sebagainya. Dalam penelitian ini, metode prototyping digunakan untuk mempermudah komunikasi dan pemenuhan kebutuhan pengguna karena fleksibilitas dan kemampuan metode ini dalam beradaptasi terhadap kebutuhan pengguna, utamanya kebutuhan fungsional yang dapat berubah seiring berjalannya pengembangan perangkat lunak.

B.6

1.4.2 PHPUnit dan ci-phpunit-test

PHPUnit adalah tools atau perangkat dan kerangka kerja untuk melakukan unit testing pada bahasa pemrograman PHP. Unit testing yang dilakukan menggunakan PHPUnit lebih berorientasi terhadap pengembang perangkat lunak, karena pengujian yang dilakukan bersifat programming, atau menggunakan logika-logika programming untuk menentukan apakah perangkat lunak akan memberikan hasil yang diharapkan. Sebuah unit testing yang baik tidak hanya menguji apabila perangkat lunak akan memberikan hasil yang benar, namun juga saat perangkat lunak seharusnya menghasilkan error atau kesalahan. PHPUnit sendiri merupakan salah satu bentuk dari arsitektur kerangka kerja unit testing xUnit. Pada penelitian ini pengembang menggunakan PHPUnit versi 6 menyesuaikan versi PHP yang digunakan.

Sedangkan ci-phpunit-tests merupakan sebuah alat bantu untuk mempermudah penggunaan PHPUnit pada framework CI, dikarenakan CI memiliki fungsi-fungsi tambahan yang membuatnya berbeda dari implementasi pemrograman PHP pada umumnya. Sehingga PHPUnit perlu dirubah agar dapat melakukan unit testing pada aplikasi PHP berbasis framework CI.

1.4.3 PhpSpreadsheet

PhpSpreadsheet merupakan library php yang menyediakan seperangkat fungsi untuk membaca dan menulis ke bermacam format file spreadsheet seperti Excel atau LibreOffice Calc. PhpSpreadsheet merupakan perkembangan dari PHPExcel yang sekarang sudah tidak dikembangkan lagi.

1.5 Ikhtisar

Gambaran umum dari dokumen SRS ini adalah mengenai spesifikasi kebutuhan software yang akan dibuat. Selain itu

dokumen SRS ini juga akan meliputi tinjauan kebutuhan fungsional dan non-fungsional dari aplikasi yang dibuat.

2. Deskripsi Keseluruhan

2.1 Kegunaan Produk

Manfaat utama dari perangkat lunak yang akan dibuat adalah untuk menampilkan kondisi bangunan terkini dari seluruh bangunan yang ada di ITS dan sudah terdaftar pada inventaris ITS. Sistem informasi yang dihasilkan akan dapat menampilkan data-data terkait gedung dan bangunan yang sudah ada. Sistem juga dirancang untuk dapat membantu pengguna untuk mengajukan dan memantau perawatan gedung serta melihat data-data gedung terkait hubungannya dengan pengelola gedung.

Dengan manfaat tersebut diharapkan sistem informasi dapat membantu pihak-pihak Subdirektorat Implementasi Rencana Induk ITS, Wakil Rektor II, bagian Sarana dan Prasarana, serta unit-unit fakultas dan jurusan di ITS.

2.2 Karakteristik Pengguna

No	Pengguna	Tanggung Jawab	Fungsi
1	Admin	<p>Melakukan manajemen data pengguna aplikasi</p> <p>Melakukan pengelolaan terhadap seluruh data, termasuk data gedung, renovasi, dan pekerjaan atas</p>	<p>Dapat melakukan fungsi CRUD pada data pengguna aplikasi</p> <p>Dapat melakukan fungsi reset password apabila pengguna lupa</p>

B.8

		permintaan pengguna lain	
2	SIMRI	<p>Melihat dan mengubah data gedung yang telah ada serta mengisikan data gedung</p> <p>Melihat dan membuat daftar pekerjaan renovasi gedung</p> <p>Menetapkan anggaran renovasi gedung</p> <p>Melakukan pengajuan renovasi gedung</p>	<p>Dapat melakukan fungsi CRUD pada data gedung ITS</p> <p>Dapat melakukan fungsi CRUD pada data pekerjaan renovasi gedung</p> <p>Dapat melakukan fungsi CR pada data anggaran renovasi gedung</p> <p>Dapat melakukan fungsi CR pada data renovasi gedung</p>
3	WR2	Melakukan persetujuan anggaran renovasi gedung	Dapat melakukan RU pada data anggaran renovasi gedung
4	SARPRAS	Melakukan pengecekan/validasi pekerjaan renovasi	Dapat melakukan RU pada data

			pekerjaan gedung
5	Unit Fakultas/Jurusan	Melakukan pengajuan renovasi gedung	Dapat melakukan fungsi CR pada data renovasi gedung

2.3 Batasan

Ruang lingkup dokumen SRS ini adalah mengenai deskripsi kebutuhan fungsional dan non-fungsional dari aplikasi yang akan dibuat, diantaranya adalah: penggunaan (usability), keandalan (reliability), keamanan (security) aplikasi dan kinerja (performance) aplikasi berbasis web.

Sistem informasi yang dibuat akan mampu menampilkan data gedung di ITS secara tepat dan terkini, sehingga akan membantu pihak SIMRI dalam melakukan pemantauan kondisi gedung. Sistem informasi juga dapat menampilkan luas keseluruhan bangunan pada lahan ITS, beserta lokasinya pada peta yang juga disediakan oleh aplikasi untuk mempermudah visualisasi dari posisi gedung. Sistem informasi yang dikembangkan juga didesain untuk menampilkan renovasi gedung yang sudah selesai, renovasi yang sedang dikerjakan, maupun untuk melakukan pengajuan dan persetujuan renovasi gedung.

2.4 Asumsi dan Ketergantungan

Perangkat lunak tidak memiliki ketergantungan keras terhadap perangkat lunak maupun sistem lain, namun data yang terkandung dalam perangkat lunak memiliki ketergantungan lunak dengan arsip data hardcopy yang dimiliki oleh pihak pengelola gedung di ITS. Penggunaan perangkat lunak dengan asumsi bahwa data yang ada telah mengikuti data pengelola gedung terkini dan akan terus dimutakhirkan oleh pengguna –

B.10

pengguna yang memiliki hak akses terhadap data gedung yang ada. Sebagai contoh untuk data gedung apabila terjadi perubahan pada jumlah lantai, luas bangunan, dikarenakan renovasi atau alasan – alasan lain dimana pengukuran dan perbaruan data dilakukan oleh pengguna dengan hak akses SIMRI.

2.5 Bagian Kebutuhan

Bagian ini mendaftarkan kebutuhan-kebutuhan akan perangkat keras dan peralatan yang digunakan untuk mengembangkan perangkat lunak.

2.5.1 Perangkat Keras

Adapun spesifikasi kebutuhan perangkat keras yang akan digunakan untuk mengembangkan aplikasi ini adalah sebagai berikut:

1. 1 (satu) unit komputer untuk programming
Nama unit : (bebas)
Processor : Core i5
RAM : 2 GB
VGA : (bebas)
HDD : 320 GB atau lebih tinggi
2. 1 (satu) unit komputer sebagai server testing aplikasi
Nama unit : (bebas)
Processor : Core i5 atau Xeon
RAM : 2 GB atau lebih tinggi
VGA : On-board 1 GB atau lebih tinggi
HDD : 500 GB atau lebih tinggi

2.5.2 Peralatan

Adapun spesifikasi kebutuhan tools yang akan digunakan untuk mengembangkan aplikasi ini adalah sebagai berikut:

1. Adobe Photoshop

Tools ini digunakan untuk mendesain layout aplikasi yang akan dibuat. Juga digunakan untuk melakukan perubahan/modifikasi pada peta dasar ITS yang telah ada. Selain itu photoshop juga digunakan untuk melakukan desain ikon perangkat lunak. Pada dasarnya photoshop digunakan untuk keperluan desain aplikasi web yang berhubungan dengan desain dan tampilan. Photoshop yang digunakan adalah minimal versi CS6.

2. ATOM Editor

Tools ini digunakan sebagai code editor untuk keperluan utama pengembangan aplikasi web. Versi yang digunakan pada saat pengembangan aplikasi adalah versi 1.23.3 dengan versi Chrome 56.0.2924.87 dan Node 7.4.0.

3. PHPunit

PHPUnit digunakan untuk melakukan unit testing pada aplikasi untuk memastikan bahwa fungsi-fungsi pada aplikasi memberikan luaran yang diharapkan oleh pengembang sebagaimana mestinya. Pengembang menggunakan versi 6 PHPUnit mengikuti versi PHP yang digunakan yakni PHP versi 7.

4. ci-phpunit-tests

Tools ini digunakan untuk mempermudah penggunaan tools PHPUnit pada framework codeigniter. Karena codeigniter menggunakan implementasi yang berbeda dari pengembangan aplikasi web menggunakan PHP pada umumnya maka pengembang memerlukan tools tambahan untuk melakukan unit testing pada aplikasi yang dikembangkan. Tools menggunakan versi 0.15.0

B.12

stable mengikuti versi codeigniter yang digunakan yaitu codeigniter versi 3.1.6.

5. Git

Tools ini digunakan sebagai version control system. Git akan ditanam (embedded) pada web-server, sehingga nantinya bisa diakses oleh tim pengembang saat dibutuhkan untuk melakukan perubahan-perubahan berupa update aplikasi atau perbaikan bug.

6. Draw.io

Tools ini digunakan untuk mendesain Entity Relationship Diagram, Use-Case Diagram, dan Swimlane Diagram pada tahap perancangan perangkat lunak. Tools merupakan aplikasi web sehingga tidak memiliki versi tertentu.

7. Phpmyadmin

Tools ini digunakan untuk manajemen database mysql. Versi yang digunakan adalah 4.7.6.

3. Kebutuhan Spesifik

Adapun fungsi-fungsi dari aplikasi sistem informasi yang akan dibuat adalah sebagai berikut:

1. Melihat, mengubah, dan menambah data gedung di ITS
2. Menampilkan progress dan melakukan perubahan renovasi gedung pada tiap renovasi
3. Mengunggah file excel daftar pekerjaan renovasi yang disetujui
4. Menampilkan daftar dan melakukan validasi pekerjaan renovasi
5. Melakukan persetujuan atau penolakan proposal renovasi beserta alokasi anggaran dana renovasi
6. Mengkalkulasi dan menampilkan luas lahan terbangun seluruh ITS

3.1 Kebutuhan Fungsional

3.1.1 Fungsi Menambah dan Mengubah Data Gedung di ITS

Fungsi “Menambah dan Mengubah Data Gedung di ITS”	
Objective	Mengubah dan menambah data gedung agar sesuai dengan kondisi pembangunan terkini
Input	<ul style="list-style-type: none"> - Data nama gedung - Data kode gedung - Data kategori gedung - Data luas gedung - Data lantai gedung - Data lokasi gedung
Process	<ul style="list-style-type: none"> - Mengambil dan menyimpan dalam database gedung - Mengambil dan menyimpan dari database terkait gedung oleh foreign key (tabel kategori gedung)
Output	Data gedung secara terinci, dalam bentuk tampilan web

B.14

Dependency	-
Actor	Pegawai SIMRI

3.1.2 Fungsi Mengajukan dan Mengubah Data Renovasi Gedung

Fungsi “Mengajukan dan Mengubah Data Renovasi Gedung”	
Objective	Melakukan pengajuan renovasi baru terhadap gedung di ITS
Input	<ul style="list-style-type: none">- Data nama gedung- Alasan renovasi gedung
Process	<ul style="list-style-type: none">- Mengambil dan menyimpan dalam database renovasi- Memberikan status ‘belum disetujui’ pada ajuan renovasi
Output	Data pengajuan renovasi untuk ditinjau ulang
Dependency	-
Actor	Pegawai SIMRI dan/atau Unit/Fakultas

3.1.3 Fungsi Persetujuan dan Penolakan Renovasi Gedung

Fungsi “Mengajukan dan Mengubah Data Renovasi Gedung”	
Objective	Melakukan pengajuan renovasi baru terhadap gedung di ITS
Input	<ul style="list-style-type: none">- Data nama gedung- Alasan renovasi gedung- Anggaran renovasi (bila disetujui)
Process	<ul style="list-style-type: none">- Mengambil dan menyimpan dalam database renovasi- Mengubah status renovasi ke ‘disetujui’ atau ‘ditolak’
Output	Data pengajuan renovasi untuk ditinjau ulang dan diberi daftar pekerjaan (apabila disetujui)
Dependency	-
Actor	Pegawai SIMRI dan/atau Unit/Fakultas

3.1.4 Fungsi Menambah dan Mengubah Pekerjaan Renovasi Gedung

Juga termasuk disini adalah fungsi pengunggahan data pekerjaan dalam bentuk excel.

Fungsi “Menambah dan Mengubah Pekerjaan Renovasi Gedung”	
Objective	Menambah dan/atau mengubah data pekerjaan renovasi untuk dilakukan pada saat renovasi gedung
Input	<ul style="list-style-type: none"> - Data gedung terkait - Data ajuan renovasi - Data pekerjaan renovasi yang harus dilakukan (dapat berupa file excel berupa daftar pekerjaan)
Process	<ul style="list-style-type: none"> - Mengambil dan menyimpan dalam database pekerjaan renovasi - Melakukan translasi file excel ke dalam format pekerjaan yang didukung pada database (apabila daftar pekerjaan dalam bentuk file excel)
Output	Daftar pekerjaan renovasi terkait renovasi yang telah disetujui pada gedung
Dependency	Data renovasi gedung yang telah disetujui
Actor	Pegawai SIMRI

3.1.5 Fungsi Melakukan Pengecekan Pekerjaan Renovasi Gedung

Fungsi “Melakukan Pengecekan Pekerjaan Renovasi Gedung”	
Objective	Melakukan pengecekan pada daftar pekerjaan renovasi yang telah dilakukan tiap harinya
Input	<ul style="list-style-type: none"> - Data renovasi gedung - Daftar pekerjaan renovasi yang sudah dilakukan

B.16

Process	<ul style="list-style-type: none">- Mengambil dan menyimpan dalam database pekerjaan renovasi- Melakukan perhitungan <u>progress renovasi gedung</u>
Output	Daftar pekerjaan renovasi yang sudah dilakukan dan <u>progress renovasi gedung</u>
Dependency	Daftar pekerjaan renovasi gedung
Actor	SARPRAS

3.1.6 Fungsi Menandai Renovasi Telah Selesai

Fungsi “Menandai Renovasi Telah Selesai”	
Objective	Melakukan pemutusan saat renovasi dianggap sudah selesai, dengan seluruh daftar pekerjaan telah dilakukan maupun belum
Input	<ul style="list-style-type: none">- Data gedung- Data renovasi gedung
Process	<ul style="list-style-type: none">- Mengambil dan menyimpan dalam database pekerjaan renovasi dan renovasi gedung- Mengubah status renovasi ‘selesai’
Output	Daftar pekerjaan renovasi terkait renovasi yang telah disetujui pada gedung
Dependency	Data renovasi gedung dan pekerjaan renovasi terkait
Actor	Pegawai SIMRI

3.1.7 Fungsi Reset Password Pengguna

Fungsi “Reset Password Pengguna”	
Objective	Melakukan reset password pengguna ke password default apabila pengguna lupa password
Input	<ul style="list-style-type: none">- Data pengguna
Process	<ul style="list-style-type: none">- Mengubah password pengguna ke password default (123456)

	- Menandai bahwa pengguna menggunakan password default
Output	Password pengguna kembali ke password default dan menunjukkan notifikasi pada saat pengguna melakukan login
Dependency	-
Actor	Admin

3.1.8 Fungsi Penggantian Password Pengguna

Fungsi “Penggantian Password Pengguna”	
Objective	Mengubah/memperbarui password pengguna
Input	- Password baru pengguna
Process	- Mengambil dan menyimpan dalam database pengguna
Output	Notifikasi saat password berhasil diubah
Dependency	-
Actor	Pegawai SIMRI, WR 2, SARPRAS, dan Unit/Fakultas

3.2 Keamanan

Aplikasi menggunakan framework codeigniter dan menyamarkan ID dari data-data pengguna, gedung, dan sebagainya sehingga tidak terlihat pada bilah alamat browser agar tidak diketahui sumber pada database. Aplikasi juga selalu melakukan pengecekan sesi login pengguna pada seluruh fungsi untuk memastikan pengguna memiliki hak akses, terkecuali pada fungsi – fungsi yang bisa diakses tanpa diperlukan login. Fungsi – fungsi yang tidak dilakukan pengecekan login hanya pada fungsi pencarian gedung dan melihat data gedung.

3.2.1 Keamanan Fungsi Perubahan Password Pengguna

Data yang masuk akan dienkripsi menggunakan enkripsi hashing MD5 untuk menjamin enkripsi tidak bisa didekripsi lagi.

3.3 Keandalan

Keandalan aplikasi menjelaskan mengenai keandalan aplikasi berupa ketersediaan data dan/atau akurasi data terhadap arsip yang dimiliki pihak pengelola dan kondisi terkini bangunan di ITS. Keandalan aplikasi dijelaskan pada tiap fungsi yang bersangkutan.

3.3.1 Keandalan Fungsi Menambah dan Mengubah Data Gedung

Data rinci gedung di ITS harus selalu tersedia 24/7, dan hanya non-aktif ketika dilakukan maintenance data oleh tim Admin untuk melakukan rekapitulasi data online dan memastikan data sesuai dengan arsip yang dimiliki oleh SIMRI. Maintenance dilakukan agar data gedung yang tercantum pada aplikasi akurat dengan kondisi terkini dan dapat dipertanggungjawabkan oleh pihak pengelola gedung.

3.3.2 Keandalan Fungsi Mengajukan dan Mengubah Data Renovasi Gedung

Keandalan data yang dicantumkan pada bagian ini juga termasuk fungsi persetujuan renovasi beserta anggaran dana, penolakan renovasi beserta alasan, dan fungsi renovasi selesai dikarenakan fungsi – fungsi yang saling terkait dalam memanipulasi data yang sama (data renovasi gedung).

Data pengajuan renovasi, beserta renovasi yang telah disetujui, sedang berjalan, renovasi ditolak, dan renovasi yang telah selesai selalu tersedia dan dapat dibuka pada aplikasi web 24/7. Pihak pengelola gedung juga selalu melakukan pengecekan sehingga data yang dicantumkan akurat dengan arsip dan surat – surat keputusan yang ada.

3.3.3 Keandalan Fungsi Menambah dan Mengubah Pekerjaan Renovasi Gedung

Keandalan data disini juga mencakup fungsi pengecekan pekerjaan renovasi karena menggunakan data yang sama, yaitu data pekerjaan renovasi, hanya dilakukan oleh dua hak akses pengguna yang berbeda.

Data pekerjaan renovasi selalu tersedia pada aplikasi, dan dapat dilihat dan diubah oleh pengguna sesuai hak akses masing – masing, selama data renovasi dan pekerjaan tidak dihapus oleh pengguna Admin atas alasan tertentu.

3.3.4 Keandalan Fungsi Reset Password Pengguna dan Mengubah Password

Data login pengguna (termasuk username, password, hak akses, dan lain – lain) dapat diakses 24/7. Karena data digunakan sebagai dasar login pengguna maka data tidak akan dimatikan meskipun pada saat maintenance sistem.

3.4 Kinerja

Kinerja menjelaskan mengenai kehandalan data ataupun proses berupa waktu respon dan kapasitas.

3.4.1 Kinerja Fungsi Menambah dan Mengubah Data Gedung

3.4.1.1 Waktu Respon

DBMS yang digunakan mampu menangani akses pengambilan dan penulisan data yang dilakukan oleh sedikitnya 10 pengguna dalam satu waktu sekaligus.

3.4.1.2 Kapasitas

DBMS yang digunakan mampu untuk menampung minimum 1000 baris data per tabel.

B.20

3.4.2 Kinerja Fungsi Terkait Renovasi Gedung

Disini fungsi – fungsi yang terkait dengan data renovasi gedung termasuk mengajukan dan mengubah data renovasi, fungsi persetujuan dan penolakan renovasi, serta fungsi menandai renovasi telah selesai.

3.4.2.1 Waktu Respon

DBMS yang digunakan mampu menangani akses data yang dilakukan oleh sedikitnya 10 pengguna dalam keadaan login pada satu waktu.

3.4.2.2 Kapasitas

DBMS yang digunakan mampu untuk menampung minimum 1000 baris data pada tabel renovasi gedung.

3.4.3 Kinerja Fungsi Pekerjaan Renovasi

Fungsi – fungsi yang terkait dengan data pekerjaan renovasi adalah fungsi menambahkan dan mengubah pekerjaan renovasi, dan fungsi pengecekan pekerjaan renovasi. Fungsi untuk menambahkan pekerjaan renovasi dipisah tersendiri untuk fungsi mengunggah file excel daftar pekerjaan renovasi.

3.4.3.1 Waktu Respon

DBMS yang digunakan mampu menangani akses data yang dilakukan oleh sedikitnya 10 pengguna dalam satu waktu sekaligus. Namun apabila menggunakan fungsi unggah file excel maka waktu respon aplikasi akan sedikit bertambah, meskipun perbedaannya tidak signifikan.

3.4.3.2 Kapasitas

DBMS yang digunakan mampu untuk menampung minimum 1 juta baris data pada tabel pekerjaan renovasi gedung.